NBSIR 76-994

A Numerical Solution of the Time Dependent Partial Differential Equations which describe a one-dimensional, Laminar, Premixed Flame

R. L. Brown

Institute for Materials Research
National Bureau of Standards
Washington, D. C. 20234

January 1976

Final

U. S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NBSIR 76-994

A NUMERICAL SOLUTION OF THE TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS WHICH DESCRIBE A ONE-DIMENSIONAL, LAMINAR, PREMIXED FLAME

R. L. Brown

Institute for Materials Research
National Bureau of Standards
Washington, D. C. 20234

January 1976

Final

A NUMERICAL SOLUTION OF THE TIME DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS WHICH DESCRIBE A ONE-DIMENSIONAL, LAMINAR, PREMIXED FLAME

by

R. L. Brown

Institute for Materials Research

National Bureau of Standards

Washington, D. C. 20234

ABSTRACT

The set of time dependent, parabolic differential equations, which describe the physical and chemical processes in a one-dimensional, laminar, premixed flame is solved by adapting a solution procedure originally developed to solve the two-dimensional steady state boundary layer equations. The flame equations are integrated by an implicit method until the steady state is reached. This corresponds to a flame propagating steadily through a mixture of combustible gases. By a suitable choice of boundary conditions, it is also possible to model a flame which is stabilized on a burner. Solution of the flame equations yields the concentration profiles of the different chemical species as well as the temperature profile. From these one can also calculate the production rates of each species, the rate of each chemical reaction, and the heat release rate at each point in the flame. The velocity of the freely propagating flame can be calculated from the integrals over the whole flame zone of any of the species production rates. The model incorporates realistic thermodynamic data and transport property data that are functions of both temperature and concentration. A complete documentation of the computer program which accomplishes the integration is presented.

TABLE OF CONTENTS

I. INTRODUCTION

The factors which determine how fast a flame will move through a mixture of combustible gases have been known for many years. Flames of practical interest are generated by chemical reactions involving free radicals. The radical reactions usually have high activation energies and thus are rapid only at elevated temperatures. A flame propagates because the cold gases in its path are raised in temperature by thermal conduction and infused with radicals by mass diffusion from the flame front. It is a simple matter to set up equations for the conservation of energy and mass which give the time dependence of the temperature and concentration of chemical species at different positions in the flame. These equations contain transport parameters and chemical rate constants. The transport parameters consist of mass diffusion coefficients for each chemical species in the mixture and the thermal conductivity of the mixture. The kinetic theory of gases provides the functional dependence of these transport properties on the composition and temperature of the mixture. In addition, it also requires a knowledge of binary diffusion coefficients between pairs of species and the thermal conductivities of the pure species. Although the mathematical expressions for the transport properties are very complex, this constitutes no impediment to the solution of the flame equations; their changes with time are slow and do not require calculation at every integration step. Furthermore, the kinetic theory is sufficiently accurate, and our experimental and theoretical knowledge of diffusion coefficients and thermal conductivities sufficiently good that the transport phenomena impose no major barrier to a flame modeling calculation.

The significant barrier arises from a lack of accurate rate constants and until the last few years in efficient numerical techniques for solving the flame equations. While the rate constant problem remains, it is now possible to solve the flame equations on a high speed digital computer in times ranging from a few seconds to several minutes depending on the complexity of the flame. The method used in the present approach was originally developed by Patankar and Spalding[1] to solve the two-dimensional steady-state boundary layer equations. One of the two

spatial variables in that problem becomes time in the flame equations, while the other represents the distance through the flame fromt. The equations for the two systems are otherwise the same.

In reading this report, it is desirable to begin with Section II. This gives a brief discussion of the method used to solve the flame equations. Appendices A and B should then be read. These give a simple derivation of the one-dimensional flame equations and show how the steady-state flame velocity is calculated from the solutions of the equations. These equations contain the diffusion velocities of the various species. The expressions used for these are derived in Appendix C. We have considered here only the diffusion arising from a concentration gradient. That caused by a thermal gradient has been neglected. Appendix D, which shows how the thermal conductivity of the flame was calculated can be skipped on a first reading. Appendices E and F show how the flame equations are transformed to a new spatial variable. This is done in two steps. The first is the Von Mises transformation which eliminates the equation for overall mass conservation. This removes the explicit dependence on the flame velocity in the remaining equations. Next, a transformation devised by Patankar and Spalding is used to convert the equations into the form of the boundary layer equations. Appendix G discusses the assumptions underlying the unity Lewis number approximation and can be skipped. Appendices H through K show how the equations are written in finite-difference form, how the various source terms are calculated, the way the grid size is controlled during the integration of the equations, and how the boundary conditions are incorporated into the calculation. The last three Appendices can be skipped.

After this, one can begin with Section III, which gives a detailed discussion of how the program works. Throughout this section, references are made to relevent parts of the Appendices for detailed discussions of the various calculations.

A test case is presented in Section IV. For this a H_2-Br_2 flame was used. This shows the form of the input data and the type of output that the program yields.

II. METHOD USED TO SOLVE THE FLAME EQUATIONS

In this section we present an outline of the method used to solve the flame equations. The details are presented in the Appendices.

The equations which describe a one-dimensional pre-mixed laminar flame can be written in the form of the classical diffusion equation.

$$\frac{\partial C_j}{\partial t} = \frac{\partial}{\partial x}\left(\tau_j \frac{\partial C_j}{\partial x}\right) + S_j \qquad j = 1,\ldots\ldots\ldots,N \qquad (1)$$

C_j is the concentration of the j'th species. τ_j is a diffusion parameter and S_j is the chemical source term for j. Both τ_j and S_j are functions (presumed to be known) of the concentrations of the various species. The independent variables are the time t and a distance x. x is actually a transformed spatial variable which is roughly proportional to the distance y which would be measured in the laboratory. ($x = \psi$, see Appendix E.) There is one of these equations for each species in the flame, and also a similar equation for the temperature or the enthalpy. The derivation of these equations is given in Appendices A and E.

To solve these equations we set up a two-dimensional grid for the independent variables as shown in Fig. 1. The subscript on C in the figure now denotes a particular grid point on the x axis and is not a species index. The primes denote the values of the C_m at the time t'. The differential equation (1) gives the value of $\partial C/\partial t$ in terms of the derivatives of the spatial variable. Thus, at a particular grid point m we have,

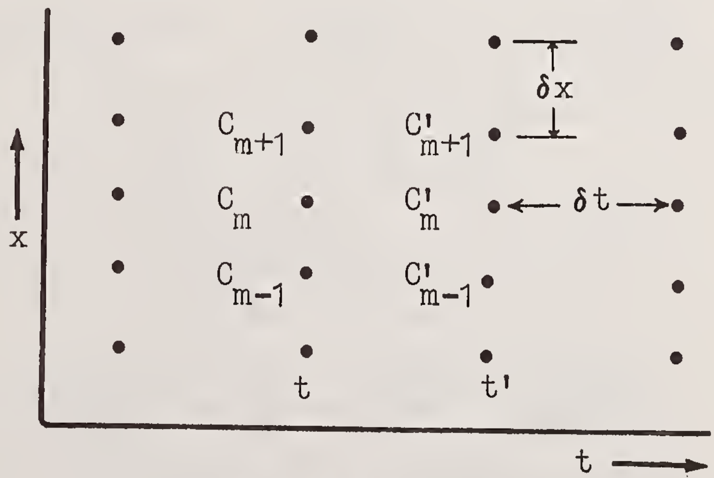


Figure 1. Grid Structure

$$\left\{\frac{\partial C}{\partial t}\right\}_m = \left\{\frac{\partial \tau}{\partial x}\right\}_m \left\{\frac{\partial C}{\partial x}\right\}_m + \tau_m \left\{\frac{\partial^2 C}{\partial x^2}\right\}_m + S_m \qquad (2)$$

To simplify the discussion, let us assume that $\partial\tau/\partial x = 0$; i.e., τ is a constant independent of x. We also will assume that τ is independent of time. This gives

$$\left\{\frac{\partial C}{\partial t}\right\}_m = \tau\left\{\frac{\partial^2 C}{\partial x^2}\right\}_m + S_m \tag{3}$$

To approximate the spatial derivative, assume that C can be represented by a quadratic function in x in the region from x_{m-1} to x_{m+1}. Thus, let

$$C = a + bx + cx^2 \qquad \text{for} \quad x_{m-1} \leq x \leq x_{m+1}$$

From this we get

$$\frac{\partial C}{\partial x} = b + 2cx; \qquad \left\{\frac{\partial C}{\partial x}\right\}_m = b + 2cx_m$$

$$\frac{\partial^2 C}{\partial x^2} = 2c = \left\{\frac{\partial^2 C}{\partial x^2}\right\}_m$$

To obtain the parameter c, we must solve the three equations

$$C_{m+1} = a + bx_{m+1} + cx_{m+1}^2$$

$$C_m = a + bx_m + cx_m^2$$

$$C_{m-1} = a + bx_{m-1} + cx_{m-1}^2$$

A little manipulation of these equations yields

$$c = \tfrac{1}{2}(C_{m-1} - 2C_m + C_{m+1})/(\delta x)^2$$

where δx is the spacing between the grid points on the x axis. This gives us an approximate expression for $\left\{\partial^2 C/\partial x^2\right\}_m$ in terms of the values of C at the three grid points m-1, m, and m+1.

$$\left\{\frac{\partial^2 C}{\partial x^2}\right\}_m \approx \frac{1}{\delta x^2}(C_{m-1} - 2C_m + C_{m+1}) \tag{4}$$

From (3) and (4) we have

$$\left\{\frac{\partial C}{\partial t}\right\}_m \approx \frac{\tau}{\delta x^2}(C_{m-1} - 2C_m + C_{m+1}) + S_m \tag{5}$$

$$\left\{\frac{\partial C}{\partial t}\right\}_m' \approx \frac{\tau}{\delta x^2}(C'_{m-1} - 2C'_m + C'_{m+1}) + S'_m \tag{6}$$

Equation (5) gives us an approximate value of $\left\{\partial C/\partial t\right\}_m$ which could be used to estimate C'_m from C_m;

$$C'_m \approx C_m + \left\{\frac{\partial C}{\partial t}\right\}_m \delta t \tag{7}$$

This is not the best estimate for C'_m. If C_m varied with time as shown in Fig. 2, one could get a value of C'_m closer to the true value by using the average of $\partial C/\partial t_m$ and $\left\{\partial C/\partial t\right\}'_m$; i.e.,

$$C'_m \approx C_m + \tfrac{1}{2}\left[\left\{\frac{\partial C}{\partial t}\right\}_m + \left\{\frac{\partial C}{\partial t}\right\}'_m\right] \delta t \tag{8}$$

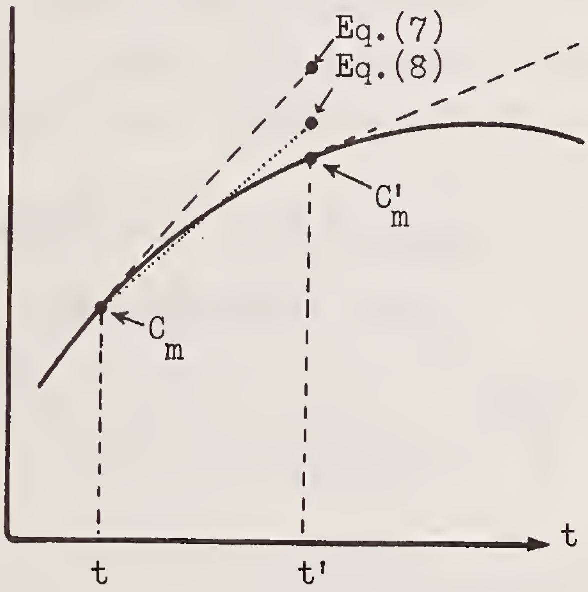


Figure 2.

To be general, let us take a weighted average of the two time derivatives,

$$\left\langle \frac{\partial C}{\partial t} \right\rangle = (1 - \lambda)\left\{\frac{\partial C}{\partial t}\right\}_m + \lambda\left\{\frac{\partial C}{\partial t}\right\}'_m$$

Our estimate for C'_m will now be

$$C'_m \approx C_m + \left\langle \frac{\partial C}{\partial t} \right\rangle \delta t$$

$$\begin{aligned} &= C_m + (1 - \lambda)\frac{\delta t\ \tau}{\delta x^2}(C_{m-1} - 2C_m + C_{m+1}) + (1 - \lambda)\delta t S_m \\ &\quad + \lambda \frac{\delta t\ \tau}{\delta x^2}(C'_{m-1} - 2C'_m + C'_{m+1}) + \lambda\ \delta t S'_m \end{aligned} \tag{10}$$

We are now confronted with the problem of determining S'_m, the value of the source term at t'. For a small change in S_m, we have

$$S'_m \approx S_m + \frac{\partial S_m}{\partial C_{1m}}(C'_{1m} - C_{1m}) + \ldots\ldots\ldots\ldots + \frac{\partial S_m}{\partial C_{Nm}}(C'_{Nm} - C_{Nm})$$

where the first subscript on C is a species index. The difficulty here is that we don't have values for the C'_{jm} until an equation like (10) is evaluated for each species j. To get around this problem, we assume that all of the partial derivatives $\partial S_m/\partial C_{jm}$ are negligible except in the case of the particular species under consideration in (10).

We then have, (dropping the species index),

$$S'_m \approx S_m + \frac{\partial S_m}{\partial C_m}(C'_m - C_m) \tag{11}$$

This is obviously not a good assumption. Nevertheless, we need only a rough estimate of S_m at t'. As t increases, we eventually reach a steady-state in which $C'_m = C_m$. Therefore S_m also stops changing with time. One could probably use the value of S_m at time t for S'_m and still reach the same steady-state. Using (11) to estimate S'_m, however, allows one to take somewhat larger time steps and consequently speeds up the integration.

Inserting (11) into (10) and rearranging, we get,

$$C'_m = A_m C'_{m-1} + A_m C'_{m+1} + B_m \tag{12}$$

where

$$A_m = A(\lambda)/D_m(\lambda)$$

$$B_m = B_m(\lambda)/D_m(\lambda)$$

$$A(\lambda) = \lambda \delta t \tau/(\delta x)^2$$

$$B_m(\lambda) = (1 - \lambda)\frac{\delta t \tau}{\delta x^2}(C_{m-1} + C_{m+1}) + \left[1 - 2(1 - \lambda)\frac{\delta t \tau}{\delta x^2} - \lambda \delta t \frac{\partial S_m}{\partial C_m}\right]C_m + S_m \delta t$$

$$D_m(\lambda) = 1 + 2\lambda\frac{\delta t \tau}{\delta x^2} - \lambda \delta t \frac{\partial S_m}{\partial C_m}$$

We are free to choose any value between 0 and 1 for the weighting parameter. If we take $\lambda = 0$, then

$$C'_m = B_m(0)/D_m(0) = \frac{\delta t \tau}{\delta x^2}(C_{m-1} - 2C_m + C_{m+1}) + S_m \delta t$$

This is an example of a so-called explicit difference equation, in which the value of C'_m is calculated from the concentrations at time t only.

When $\lambda \neq 0$ we have an implicit difference equation. This gives an equation connecting three grid points at time t'. This results in a set

of simultaneous algebraic equations (12) which must be solved. This set has the form

$$
\begin{array}{llllll}
A_1 C'_0 - C'_1 + A_1 C'_2 & & & & & = -B_1 \\
& A_2 C'_1 - C'_2 + A_2 C'_3 & & \mathbf{0} & & = -B_2 \\
& & A_3 C'_2 - C'_3 + A_3 C'_4 & & & = -B_3 \\
& & \vdots & & & \\
& \mathbf{0} & \vdots & & & \\
& & \vdots & & A_{M-1} C'_{M-2} - C'_{M-1} + A_{M-1} C'_M & = -B_{M-1}
\end{array}
$$

where M is the maximun number of grid points along the x axis. For this particular system involving three grid points, the solution of the above system of equations can be obtained without having to invert a matrix. Equation (12) can be written in the simpler form,

$$C'_m = A^*_m C'_{m+1} + B^*_m \tag{13}$$

where

$$A^*_m = \frac{A_m}{1 - A_m A^*_{m-1}}$$

$$B^*_m = \frac{A_m B^*_{m-1} + B_m}{1 - A_m A^*_{m-1}} \tag{14}$$

$$A^*_1 = A_1$$

$$B^*_1 = A_1 C'_0 + B_1$$

The values of C'_m at the first and last grid points C'_0 and C'_M are always specified by the boundary conditions for the problem. We can solve (12) by first calculating A^*_m and B^*_m for all the grid points starting from $m = 1$. It is then easy to obtain the C'_m values successively with (13) starting from C'_{M-1} and working down to C'_1.

This implicit method is what we have used to solve the flame equations. The actual form of the difference equations for the species and enthalpy used in the program is derived an Appendix H. While the methods used there to approximate the spatial derivatives are somewhat different from those used in this section, the basic solution method is the same.

III. DETAILED DESCRIPTION OF THE PROGRAM

The following is a step-by-step description of the main program and each of its subroutines. Flow diagrams for most of the routines are provided at the end of each section.

A. Description of the Main Program SPALD/3

The function of SPALD is to control the calculation. It uses various subroutines to calculate the parameters contained in the finite-difference coefficients in Eq. A40, Appendix H, p. A18. After calculating the coefficients, it then calls a routine CALC which solves the difference equations and yields values of the concentrations and enthalpy a time $t + \delta t$. It continues for as many time steps as the user specifies. No test is made for convergence.

SPALD begins by reading the following data:

1) RUNID(M), M=1,12
   Format(12A6)

   This is the run identification. It consists of 12 words, in A6 format, making 72 characters available for whatever identifying remarks the user wishes to employ.

2) HEADNG(M), M=1,126
   Format(63A1)

   This provides headings for the output profiles. There are 126 characters available. Two cards must be used. The output profiles are listed in 14 columns. The first 12 are for species concentrations and are in E9.5 format. Column 13 contains the enthalpy profile and is in E10.5 format; this extra space is used for the sign of the enthalpy which can be negative. The last column is for the temperature profile and has E9.5 format. The headings should be spaced within the 126 character range available so that they are aligned at the top of the appropriate column.

3) LH,LO,LOH,LHO2,LH2O2,LX,LHX,LX2,LO2,LH2O,LH2,LN2
   Format(14I2)

These are species indices for the H_2-O_2-N_2 flame. They should run from 1 to 12. In earlier versions of the program they were used in the subroutine which calculated the chemical source term. Now, however, they are needed only in a portion of the routine which calculates transport properties. They must be included although they are used only for the H_2-O_2 flame.

4) LD,LE,LC

Format(3I2)

These three numbers determine the number of integration steps for which certain subroutines are called at each step. For example, if LD = 20, then TRANS(1) is called for the first 20 integration steps. This means that a complete calculation of the transport properties is made at each of the first 20 steps. The second number controls the calling of SCHM1, which calculates the part of the chemical source term arising from the diffusion of the mean molecular weight. SCHM1 is called at each integration step for the first LE steps. The last number controls the reference to SENTP, the routine which calculates the enthalpy source term. This will be called at every step for the first LC steps.

5) LDELA,LDELB,LDELC

Format(3I2)

These numbers control the number of time steps for which TRANS(1), SCHM1, and SENTP are <u>not</u> called after the first LD, LE, and LC steps, respectively. For example, after calculating the transport properties for the first LD steps, the program, from this point on, will calculate them only at intervals of LDELA steps.

There are two special cases which can arise;

a) Suppose one does not wish to call SCHM1 at any time during the calculation. In most flames, the mean molecular weight changes little throughout the flame so that this part of the chemical source term can be neglected. To do this, let LE = 0 and LDELB = 0.

b) If one is considering a constant enthalpy flame, SENTP need not be called. To prevent this, let LC = 0 and LDELC = 0.

6) N,NII,NEE,OMR

Format(3I2,D5.1)

N is the number of slices into which the ω axis from 0 to 1 is divided into to form a grid. It can have a maximum value of 40. The grid points start at I = 2 and run to N + 2. (See Appendix H, p. A19.) Slices between the grid points NII and NEE have equal widths. There is a non-uniform grid spacing near the boundaries with the spacing increasing from I = NII to 2 and from NEE to N + 2 so that consecutive slices are in the fixed ratio OMR. This non-uniform spacing is used to increase the number of grid points in the region which has the largest concentration gradients. The spacing of the central grid points is given by the formula,

$$\left[\frac{\text{OMR**(NII - 1) + OMR**(N - NEE + 3) - 2.*OMR}}{\text{OMR - 1}} + \text{NEE - NII}\right]^{-1}$$

An example of this grid spacing is shown in Fig.3 for the case, N = 15, NII = 4, NEE = 14, and OMR = 1.5.

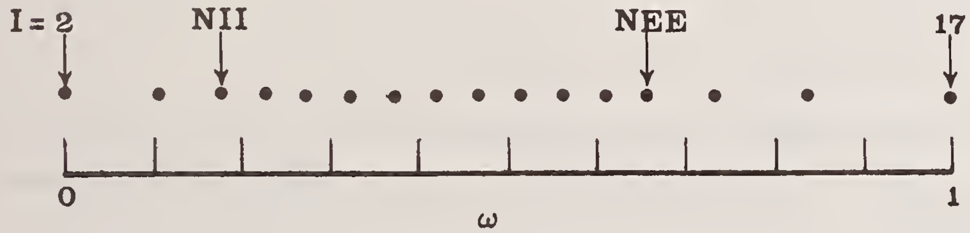


Figure 3
Sample Grid Spacing

The grid points NII and NEE also denote the positions near the hot and cold boundaries, respectively, used in the grid width control procedure. They correspond to the points HH and CC (see Appendix J).

7) JH,JRAD,JM,JMR,JT,JTR,ITER,ITER1,JENTRN,JBODYA,JBODYB

Format(11I2)

JH is the index of the enthalpy variable F(JH,I). It can have a maximum value of 13. It <u>must</u> be larger by one than the total number of species in the flame. The index of the last species will then be JHM1 = JH - 1. In the output, the number of the column will correspond

to the first index in F(J,I). The enthalpy profile, however, will always appear in column 13.

JRAD is the index of the last minor species. Minor species are always given the lowest index values. Major species start with the index JRADP1 = JRAD + 1.

JM and JMR are indices for the mean molecular weight and its reciprocal which are contained in the array FS. The mean molecular weight is FS(JM,I) and its reciprocal is FS(JMR,I). We use the values JM = 1, and JMR = 2.

JT and JTR are indices for the temperature and its reciprocal which are contained in the array FS. The temperature is FS(JT,I) and its reciprocal is FS(JTR,I). We use JT = 3 and JTR = 4.

ITER is the number of iterations used in the Newton-Raphson procedure to calculate the temperature from the enthalpy value at a grid point. ITER = 2 has been found to give satisfactory results. (See Appendix M, p. A39.)

ITER1 is not used. Set it equal to zero.

JENTRN is the index of the species which is used to calculate the entrainment rate. A major species is always used for this purpose. (See Appendix J, p. A27.)

JBODYA and JBODYB are indices for third body concentrations which are stored in the array FS. FS(JBODYA,I) is the third body concentration to be used for the reaction $H + OH + M \rightleftarrows H_2O + M$, and FS(JBODYB,I) that for the reaction $H + O_2 + M \rightleftarrows HO_2 + M$ in the H_2-O_2 flame calculations. These concentrations depend on the mixture composition and are calculated in TRANS. Normally one uses the reciprocal of the mean molecular weight for the third body concentration in the mole/kg concentration units. We let JBODYA = 5 and JBODYB = 6.

8) XI(J),J=JRADP1,JHM1
   XE(J),J=JRADP1,JHM1
   Format(5D7.3)

These are the mole fractions of all the major species at the hot and cold boundaries,respectively, at the start of the calculation. For example, consider a H_2-O_2-N_2 flame having on the cold side XE(LO2) = 0.063, XE(LH2O) = 0.0, XE(LH2) = 0.700, and XE(LN2) = 0.237. Assuming that the reaction goes to completion, we get hot side values

XI(LO2) = 0.0

XI(LH2O) = 2.*XE(LO2)/(1. - XE(LO2)) = 0.1345

XI(LH2) = (XE(LH2) - 2.*XE(LO2))/(1. - XE(LO2)) = 0.6126

XI(LN2) = XE(LN2)/(1. - XE(LO2)) = 0.2529

9) IPRFL(J),J=JRADP1,JHM1

Format(5I1)

If IPRFL(J) = 0, the initial profile for the major species J decays from the cold to the hot side of the flame. If IPRFL(J) = 1, then we have a growth profile. For the above H_2-O_2-N_2 example,

IPRFL(LO2) = 0

IPRFL(LH2O) = 1

IPRFL(LH2) = 0

IPRFL(LN2) = 1

10) TCOLD,PRESS,YWIDTH

Format(D6.0,D6.2,D5.1)

TCOLD is the temperature of the cold boundary in degrees K. PRESS is the pressure of the flame in atmospheres. YWIDTH is the approximate width of the flame front in meters. The program needs a value for this quantity to start the calculation. It uses this to calculate a starting value for the grid width parameter PEI = $\eta = \psi_C - \psi_H$. (See Appendix F, p. A11.) Although the grid width is automatically adjusted at each step in the calculation, it is best to start with a value as close as possible to the final value. We have not experimented with the effect of the choice of YWIDTH (and thus the initial PEI) on the course of the calculation.

11) HREF(J),DCP(J),ECP(J),FCP(J),WT(J),J=1,JHM1

Format(4D12.0,D10.2)

HREF(J) is the reference enthalpy value h_j^o for the species J in joules/kg. (See Appendix M, Eqs. (A78a) and (A78b), p. A37 for their definition.) DCP(J), ECP(J), and FCP(J) are the parameters d_j, e_j, and f_j in the quadratic expression used for the heat capacities of the individual species. The units are joules/kg-deg. (See Appendix M, Eq. (A76), p. A37.)

It is worth noting here that one needs values of these parameters only for the major species. The enthalpy, $h = \sum_j h_j Y_j$, so that contributions from species having small Y_j are not important in its calculation. The h_j (actually $h_j^* = M_j h_j$) also appear in the source term for the enthalpy as the products $h_j^* \partial\varphi_j/\partial\omega$. Here again the contributions from the trace species will be negligible compared to that from the major species. This is so because the values of $\partial\varphi_j/\partial\omega$ will be small when φ_j is small even though these gradients could be large on a percentage basis. Calculation time could be saved by not summing over the trace species in the enthalpy source term. The effect of neglecting these terms should be examined in future work.

WT(J) is the molecular weight of each species in kg/mole.

12) DXMIN,DXMAX,STEPS

Format(2D5.0,D6.0)

The size of the time step DX is determined from the formula DX = DXMIN + ((DXMAX - DXMIN)/STEPS)*L, where L is the number of the step and STEPS is a number comparable to the maximum number of steps to be used. The step size increases linearly from DXMIN to DXMAX when L = STEPS. We normally have DXMIN << DXMAX. Thus DX is small at the beginning of the calculation and increases as the steady-state is approached. Other step size control formulas could be investigated. Perhaps the step size could be made inversely proportional to the rate of change of the grid width PEI. As the steady-state is reached, PEI changes little from step-to-step justifying larger values of DX. At the beginning of the calculation, when the profiles are changing rapidly,

PEI will also be changing and the step size would be kept small.

13) LMAX,LPRINT

Format(2I3)

LMAX is the maximum number of steps to be used in a given calculation. LPRINT specifies the step interval at which profiles are to be printed. One page of profiles is printed every LPRINT steps. When L reaches LMAX, the complete output is printed.

14) ALPHA

Format(F3.2)

ALPHA is the parameter ν which appears in the grid control formulas Eqs. A58-60. (See Appendix J, p. A26.) We have been setting ALPHA equal to 0.1.

15) KRAT

Format(I2)

This is the number of chemical reactions. For a particular reaction, the forward and reverse reactions count as one reaction. (Note that KRATE = KRAT.) KRAT can have a maximum value of 30.

16) FRQ(K),BETA(K),EACT(K),BFLAG(K),EFLAG(K),K=1,KRATE

Format(D6.2,D7.2,D8.2,2L1)

These are the Arrehenius parameters for the <u>forward</u> rate constant of reaction K. If we express the temperature dependence of a rate constant k as $k = AT^{\beta}\exp(-E/RT)$ then

FRQ(K) = A

BETA(K) = β

EACT(K) = E/R (in ^{o}K)

k must be in units of m^3/mole and m^6/mole^2 for two and three body reactions, respectively.

BFLAG(K) = .TRUE. if BETA(K) $\neq$ 0, and EFLAG(K) = .TRUE. if EACT(K) $\neq$ 0.

17) EQA(K),EQB(K),EQC(K),BOD1(K),BOD2(K),K=1,KRATE

Format(3D11.5,2L1)

The first three quantities are the coefficients in the expansion of the free energy ΔF^{o} of each reaction as a power series in the temperature. We use

$$-\Delta F^{o} = EQA(K) + EQB(K)*T + EQC(K)*T^{2}$$

The values of these coefficients must be such that ΔF^{o} will be in units of kcal/mole.

If BOD1(K) = .TRUE., then either the forward or reverse reaction is third order. If BOD2(K) = .TRUE., then the reverse reaction is third order.

18) ITEST(M),M=1,10

Format(10I1)

When set to unity these flags produce various outputs used for checking the operation of the program. They are normally set to zero. An examination of the listing of SPALD will show what quantities are printed. They should be used only for flames with JH $\leq$ 6. With some modification of SPALD they could be used for larger values of JH.

19) INPUT

Format(I2)

When INPUT = 1, input profiles are supplied from a data file. This file must contain the concentration and enthalpy profiles and the grid width PEI. This file is taken from logical unit 45 by the statement READ(45) ((F(J,I),J=1,JH),I=1,NP3),PEI. These can be output profiles from a previous calculation. The output profiles are placed in logical unit 46 by the statement WRITE(46) ((F(J,I),J=1,JH),I=1,NP3), PEI. (NP3 equals N + 3) If the user is generating his own input profiles, they must be stored in the above order and be double precision numbers. Note that F(J,1) and F(J,NP3) are not actually used and should be given zero values. Remember that F(JH,I) is the enthalpy, and that small values of I refer to the hot side of the flame.

When INPUT = 0, the initial profiles are generated by INITL.

20) LORDER(J),J=1,JHM2

Format(14I2)

These numbers specify the order in which one wishes the species equations to be solved. Spalding has found that the efficiency of the calculation is improved by solving the species equations in order of increasing species concentration. The trace species equations are solved first, followed by the major species equations. (Note that JHM2 = JH - 2; this is the index of the next to the last major species.) The species JHM1 is calculated from the relation $\sum \varphi_j M_j = 1$, and so LORDER(JHM1) need not be specified.

As an example, consider the H_2-Br_2 flame (JH = 6) where the species have been given the indices Br = 1, H = 2, H_2 = 3, Br_2 = 4, and HBr = 5. If we wanted to solve the species equations in the order H, Br, Br_2, H_2, HBr, we would have

LORDER(1) = 2

LORDER(2) = 1

LORDER(3) = 4

LORDER(4) = 3

The equation for J = 5 will automatically be solved last.

(Note that in the test case to be presented later for which this flame was used, we solved the species equations in the order of species index, i.e., we used LORDER(J) = 1, 2, 3, 4.)

21) SPECIE(J),J=1,JHM2

Format(14L1)

If SPECIE(J) = .FALSE., then the equation for that J is not solved.

22) REACT(K),K=1,KRATE

Format(30L1)

If REACT(K) = .FALSE., then both the forward and reverse rate constants for reaction K are set to zero.

23) FFLAG(K),K=1,KRATE

RFLAG(K),K=1,KRATE

Format(30L1)

If FFLAG(K) = .FALSE., then the forward rate constant for reaction K is set to zero; if RFLAG(K) = .FALSE., then the reverse rate constant is set to zero.

24) LA(K),LB(K),LRA(K),LRB(K),K=1,KRATE
Format(4I2)

These numbers specify the reaction mechanism to be used. Consider the H_2-Br_2 flame as an example. We use the reactions,

K = 1 $\quad Br_2 + M \rightleftarrows Br + Br + M$

K = 2 $\quad H_2 + M \rightleftarrows H + H + M$

K = 3 $\quad Br + H_2 \rightleftarrows HBr + H$

K = 4 $\quad H + Br_2 \rightleftarrows HBr + Br$

For a species index assignment Br = 1, H = 2, H_2 = 3, Br_2 = 4, HBr = 5, we have the following assignment for LA, LB, LRA, and LRB:

K	LA(K)	LB(K)	LRA(K)	LRB(K)
1	4	0	1	1
2	3	0	2	2
3	1	3	5	2
4	2	4	5	1

We see that the indices LA(K) and LB(K) refer to the species entering into the forward reaction, while LRA(K) and LRB(K) pertain to those contained in the reverse reaction. There is space for two species on each side of a reaction. The third body M is not counted as a species. In cases where there is only one species aside from M involved in the reaction, one must <u>always</u> set either LB(K) or LRB(K) equal to zero. The index of the one species present should be assigned to LA(K) or LRA(K). For example, if we had written reaction 1 as

$Br + Br + M \rightleftarrows Br_2 + M$

then the assignment would have been,

LA(1) = 1, LB(1) = 1, LRA(1) = 4, LRB(1) = 0.

25) FLAG(I),I=1,20

Format(30L1)

These flags provide a series of options regarding the calculation of transport properties, source terms, and the nature of the output.

FLAG(1) = .FALSE. The complete calculation of the transport parameters is made.

= .TRUE. Constant and equal transport parameters are used. These are contained in the array PREF and are assigned values in a DATA statement in SPALD, line SPD00350. Constant, but non-equal values could be used by changing this data statement.

FLAG(2) = .FALSE. If the flame contains a buffer gas like N_2 in a low temperature H_2-O_2-N_2 flame, its mass fraction can be considered to remain constant throughout the flame. This buffer will be assigned the species index JHM1. The relation between the mass fractions gives

$\sum_{J=1}^{JHM3} Y_J + Y_{JHM2} + Y_{JHM1} = 1$.

When Y_{JHM1} is the buffer mass fraction and is kept constant, Y_{JHM2} can be calculated from this equation and its species equation need not be solved. Therefore when this option is used one must also have SPECIE(JHM2) = .FALSE..

= .TRUE. The buffer concentration is allowed to vary. This option must be used for flames containing no buffer species.

FLAG(3) = .FALSE. The pressure equals 1 atmosphere.

= .TRUE. The pressure is different from 1 atmosphere.

FLAG(4) (Not used)

FLAG(5) = .FALSE. The enthalpy source term is not calculated. This option can also be achieved by using LC = 0 and LDELC = 0.

= .TRUE. The enthalpy source term is calculated.

FLAG(6) (Not used)

FLAG(7) = .FALSE. The transport properties for the ozone decomposition flame are not calculated.

= .TRUE. The transport properties for the ozone flame are calculated.

FLAG(8) = .FALSE. The transport properties for the hydrogen-bromine flame are not calculated.

= .TRUE. The transport properties for the hydrogen-bromine flame are calculated.

FLAG(9) = .FALSE. The rate constants as a function of distance through the flame are not printed in the final output.

= .TRUE. The rate constants are printed.

FLAG(10) = .FALSE. The rates of each reaction at each point in the flame are not printed in the final output.

= .TRUE. The reaction rates are printed.

FLAG(11) = .FALSE. Heat release rates for individual reactions at each point in the flame are not printed in the final output.

= .TRUE. Heat release rates for the reactions are printed.

FLAG(12 through 20) (Not used)

This completes the input data requirements for SPALD.

After reading this data SPALD calls the subroutine SWITCH. Its function is to set three flags, G1, G2, and G3, which describe the nature of each reaction. These flags are then used in the calculation of the chemical source term by subroutine SCHM2.

FLAG(1) is then tested. If false the routine DIFUSE is called. This calculates binary diffusion coefficients for the H_2-O_2-N_2 flame from Lennard-Jones parameters. The values of these parameters are contained in a DATA statement in DIFUSE.

The index JSTOP is set equal to JHM3. FLAG(2) is then tested; if true, then the buffer gas will vary and JSTOP is equated to JHM2.

The flag INPUT is tested. If INPUT = 1, profiles are read from a file located in logical unit 45.

The quantities L, LCOUNT, X, and DX are initialized. L is the number of the step, LCOUNT is the number of steps taken since the last profile print-out, X is the time variable, and DX is the time step size.

The grid structure is calculated by a call to OMEGA.

The flag INPUT is again tested and if equal to zero INITL is called to calculate the initial profiles.

After these preliminaries, the point 706 is reached to which the program returns after every integration step. This therefore represents the beginning of each step. TRANS(1) is then called if a complete transport calculation is to be made, otherwise TRANS(0) is called. TRANS(0) calculates the temperature at each grid point from the enthalpy and stores it in FS(JT,I). TRANS(1) in addition calculates the transport parameter array PREF(J,I).

The flag INPUT is tested once again and, if zero, an initial value of PEI is determined by calling ZCALC with PEI = 1.0. This routine produces a value of Y(I), the laboratory spatial variable, at the cold boundary grid point NP2. PEI is then taken to be the ratio YWIDTH/Y(NP2). To see the rationale behind this, refer to Eq. (A70), Appendix L. This formula gives for Y(NP2)

$$Y(NP2) = 2\eta \sum_{i=2}^{N+2} \frac{\omega_{i+1} - \omega_i}{\rho_{i+1} - \rho_i}$$

The quantities ω_i were calculated in OMEGA, and the ρ_i were calculated in TRANS(1). We want to have a value of η (=PEI) such that Y(NP2) = YWIDTH. By calling ZCALC with PEI = 1.0, we get the quantity

$$2 \sum_{i=2}^{N+2} \frac{\omega_{i+1} - \omega_i}{\rho_{i+1} - \rho_i}$$

Dividing YWIDTH by this gives a value of PEI which makes Y(NP2) = YWIDTH.

SPALD next calculates the quantities PEID2 = $\frac{1}{2}\eta$, PEIOMD(I) = $\eta(\omega_{i+1} - \omega_i)$, and PEIOM2(I) = $\frac{1}{2}\eta(\omega_{i+1} - \omega_{i-1})$.

RATCN is then called to calculate the values of the rate constants at each point in the flame.

To calculate the entrainment rates $\dot{m}_C$ and $\dot{m}_H$ (see Appendix J, p. A26), it is necessary to have a value of R(JENTRN,I); this is the production rate of species JENTRN in moles/kg-s. Since this is normally calculated by SCHM2 later in the program, on the first integration step it is necessary to call SCHM2(JENTRN) to calculate the production rate for this species.

SPALD then evaluates the finite-difference coefficients A(J,I), B(J,I), and D(J,I) (minus the source term). These are the quantities A', B', and D + $2S_F$ of Eq. (A40), Appendix H, p. A18.

If desired, the chemical source term arising from the diffusion of the mean molecular wieght is then calculated by a call to SCHM1.

The finite-difference coefficient C(J,I) (minus the source term) is then calculated. This is the quantity C' - $2S_P$ of Eq. (A40).

The chemical source terms S_P and S_F are then calculated for each species by a call to SCHM2(J). Also calculated by this routine are the production rates R(J,I). Following this, C(J,I) = C' and D(J,I) = D are calculated.

New species concentrations for time t + δt (i.e., X + DX) are then calculated by calling CALC(J) in the order specified by the numbers LORDER(J).

Next, the hot boundary condition $\{\partial\varphi/\partial\omega\}_H = 0$ is set by equating the concentration at the hot boundary point to the value at the adjacent grid point; i.e., we set F(J,2) = F(J,3).

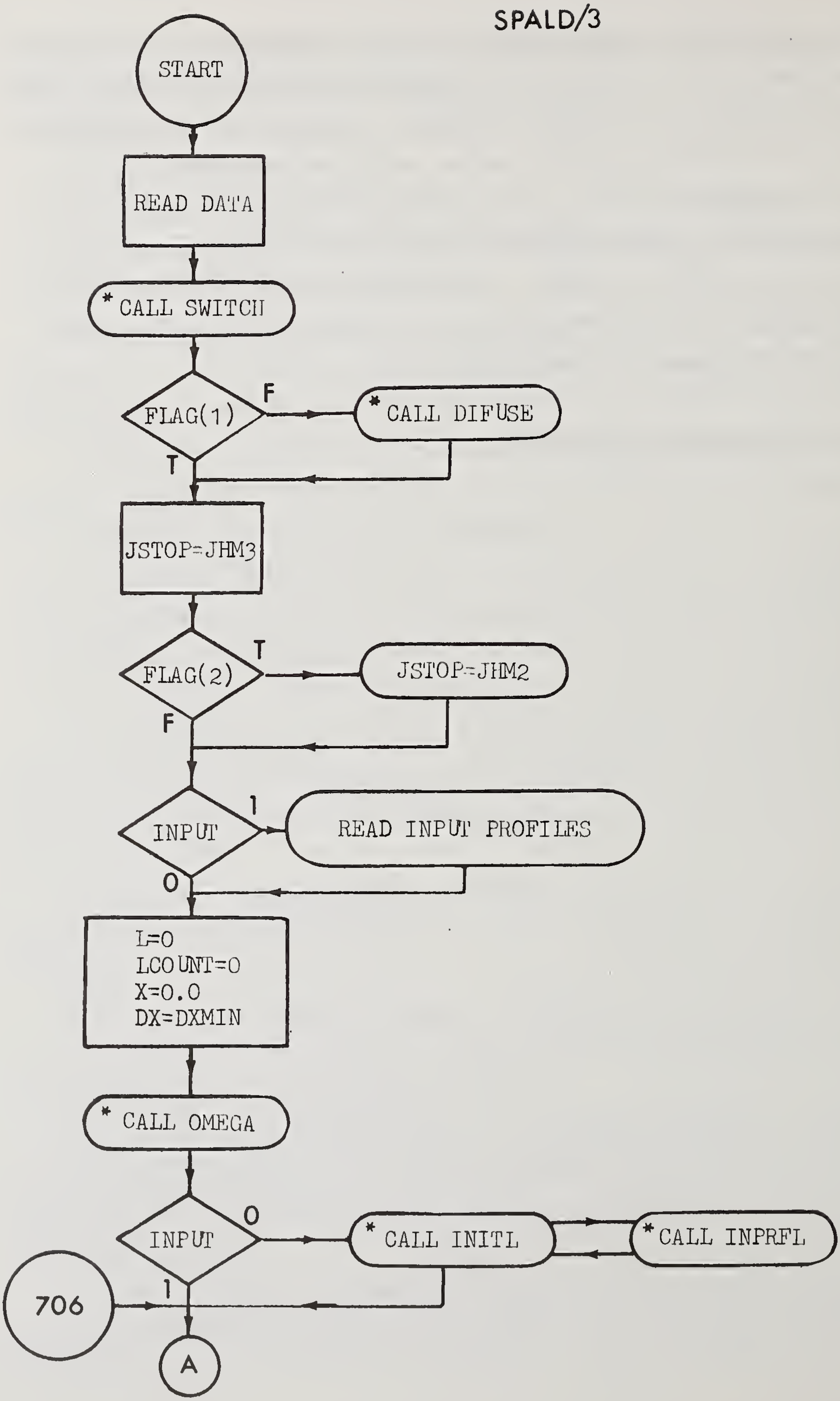
The value of the concentration of the JHM1 species is then calculated from the relation $\sum_j \varphi_j M_j = 1$; if the buffer concentration is being kept constant, then the concentration of the JHM2 species is calculated from this relation.

If the enthalpy is allowed to vary, its source term is calculated by a call to SENTP and its new value at t + δt is calculated by calling CALC(JH).

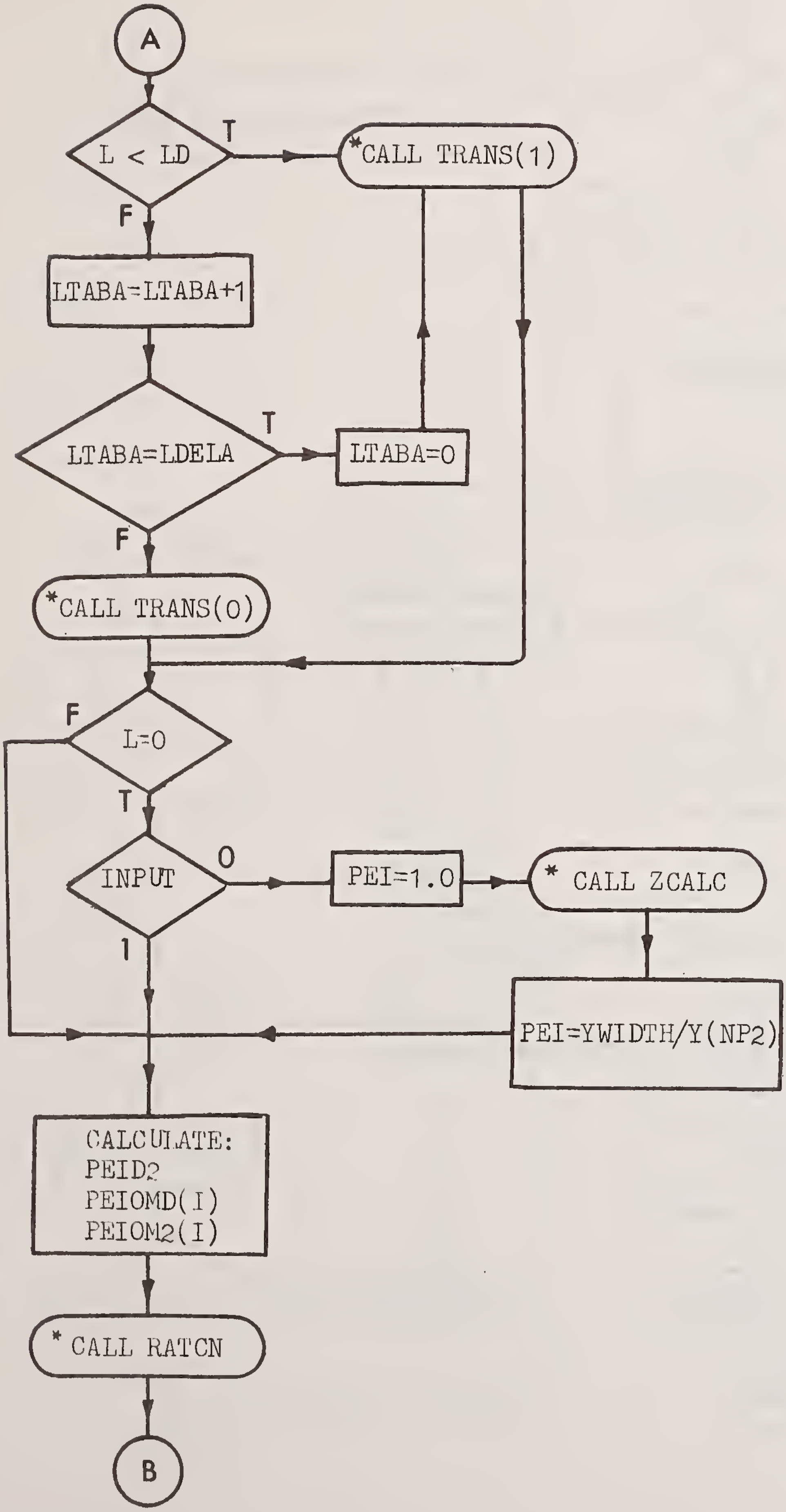
The step counters L and LCOUNT are then incremented. If an output point has been reached, flame velocities are calculated for each species by a call to ENTRN(J,1). These velocities should be the same for each species. In practice, this will happen only for the major species (see Eq. (A25) Appendix F, p. A12). The laboratory distances Y(I) are then calculated by calling ZCALC and the partial results are printed out by a call to OUTPUT. When L = LMAX, the call to OUTPUT produces a complete print out. In addition, the profiles for concentrations and enthalpy are stored in logical unit 46.

If L is not equal to LMAX, then the program returns to point 706 and the calculations are repeated.

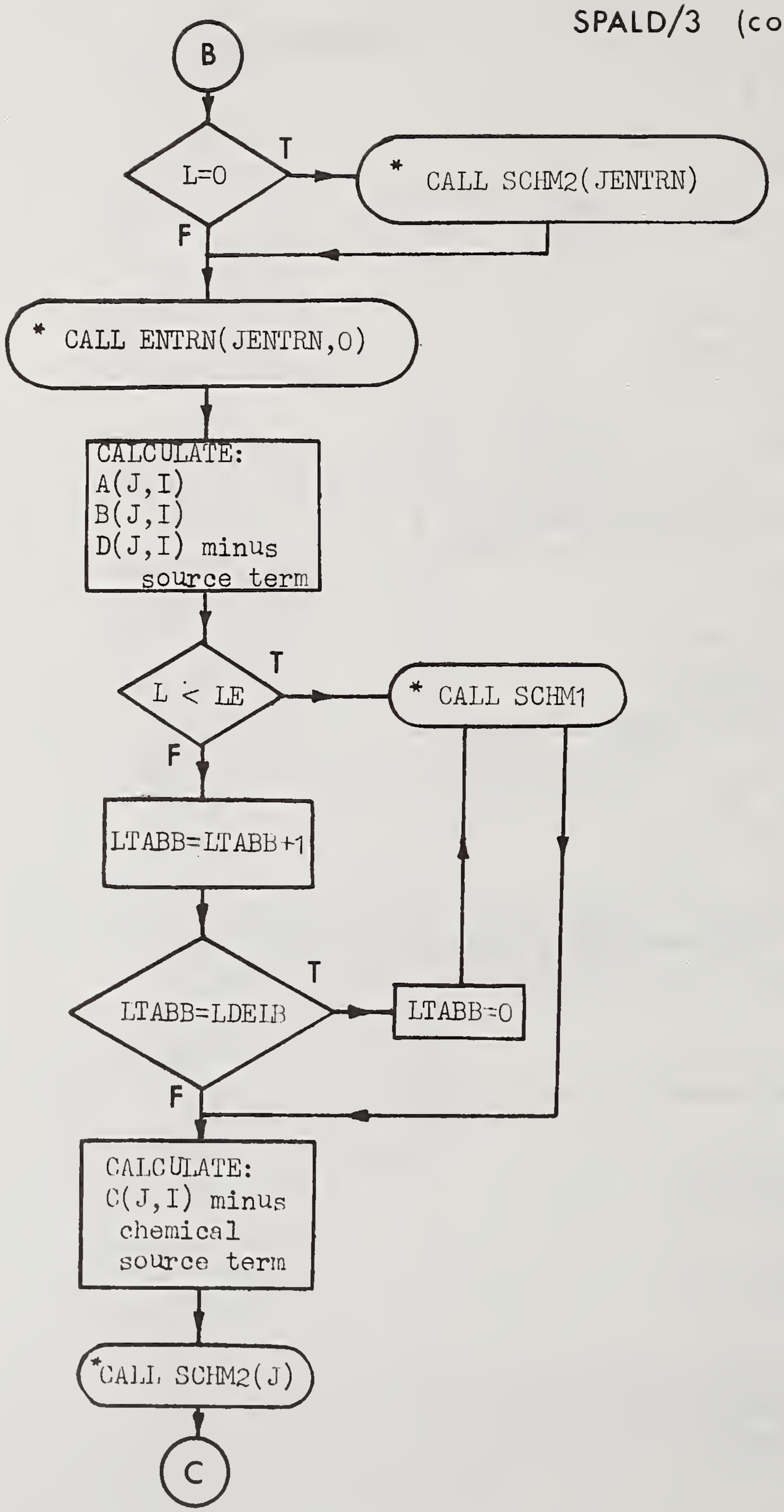
SPALD/3


START
READ DATA
* CALL SWITCH
FLAG(1)
F
* CALL DIFUSE
T
JSTOP=JHM3
FLAG(2)
T
JSTOP=JHM2
F
INPUT
1
READ INPUT PROFILES
0
L=0
LCOUNT=0
X=0.0
DX=DXMIN
* CALL OMEGA
INPUT
0
* CALL INITL
* CALL INPRFL
1
706
A

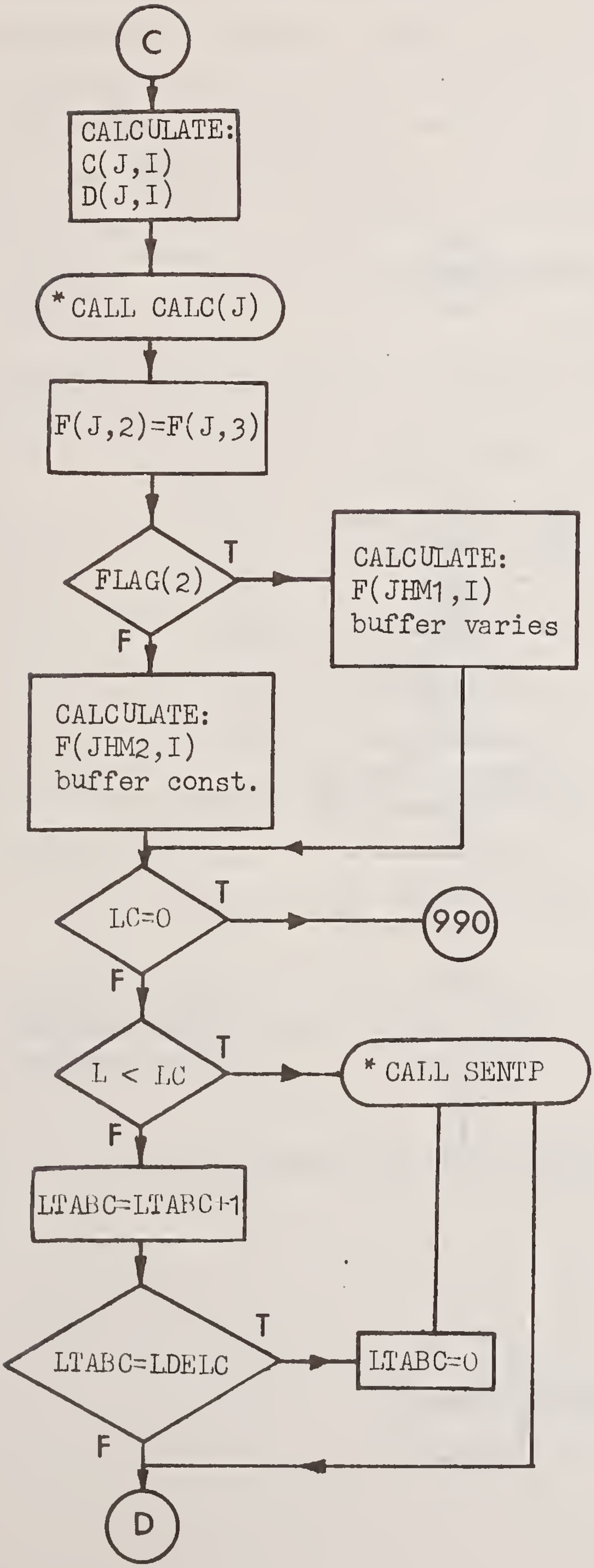
SPALD/3 (cont.)


A
L < LD
T
F
*CALL TRANS(1)
LTABA=LTABA+1
LTABA=LDELA
T
F
LTABA=0
*CALL TRANS(0)
L=0
F
T
INPUT
0
1
PEI=1.0
* CALL ZCALC
PEI=YWIDTH/Y(NP2)
CALCULATE:
PEID2
PEIOMD(I)
PEIOM2(I)
* CALL RATCN
B

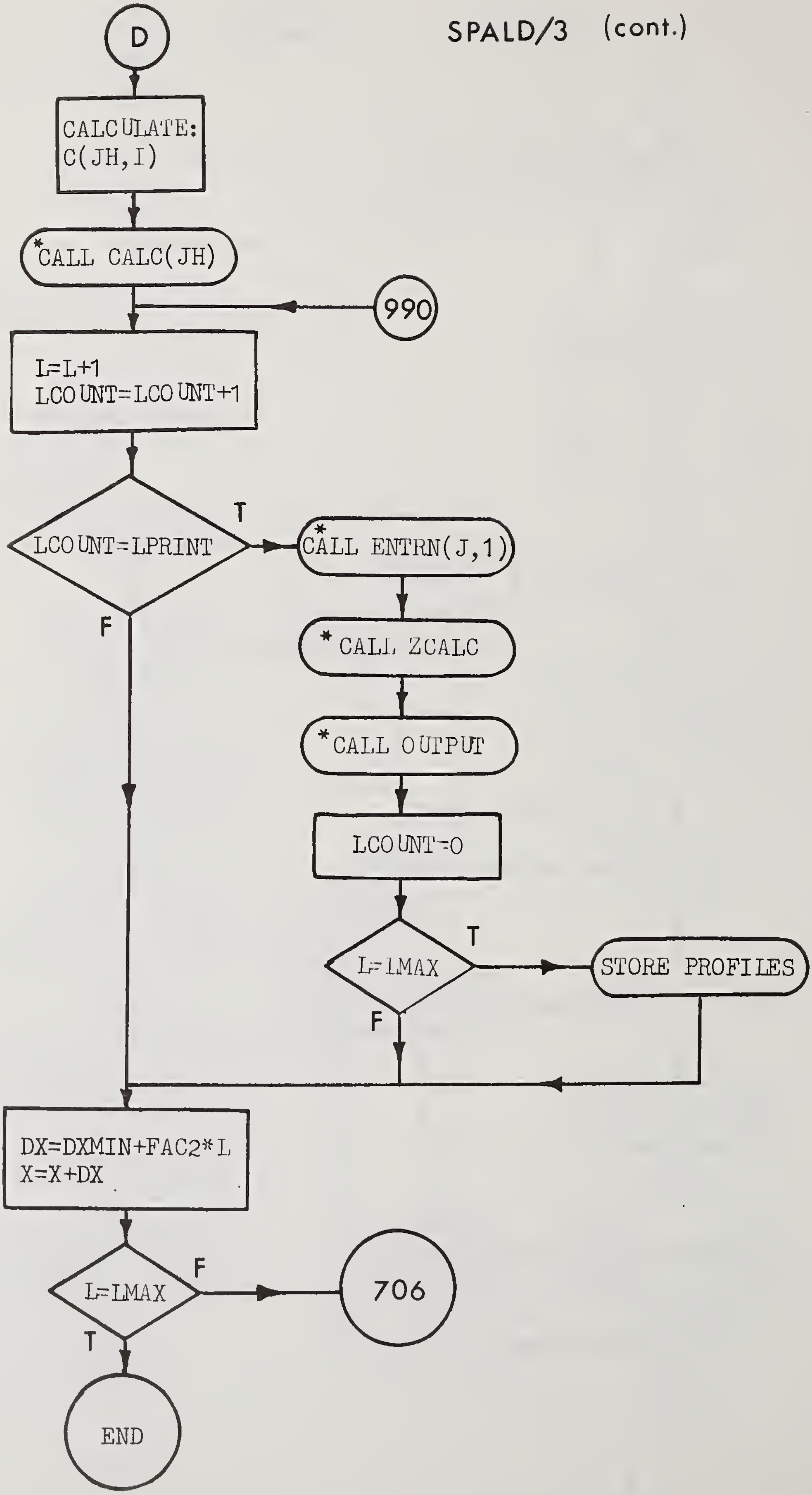
SPALD/3 (cont.)


B
L=0
T
F
* CALL SCHM2(JENTRN)
* CALL ENTRN(JENTRN,0)
CALCULATE:
A(J,I)
B(J,I)
D(J,I) minus
source term
L < LE
T
F
* CALL SCHM1
LTABB=LTABB+1
LTABB=LDELB
T
F
LTABB=0
CALCULATE:
C(J,I) minus
chemical
source term
*CALL SCHM2(J)
C

SPALD/3 (cont.)


C
CALCULATE:
C(J,I)
D(J,I)
* CALL CALC(J)
F(J,2)=F(J,3)
FLAG(2)
T
F
CALCULATE:
F(JHM1,I)
buffer varies
CALCULATE:
F(JHM2,I)
buffer const.
LC=0
990
L < LC
* CALL SENTP
LTABC=LTABC+1
LTABC=LDELC
LTABC=0
D

SPALD/3 (cont.)


D
CALCULATE:
C(JH,I)
*CALL CALC(JH)
990
L=L+1
LCOUNT=LCOUNT+1
LCOUNT=LPRINT
T
F
*CALL ENTRN(J,1)
*CALL ZCALC
*CALL OUTPUT
LCOUNT=0
L=LMAX
T
F
STORE PROFILES
DX=DXMIN+FAC2*L
X=X+DX
L=LMAX
F
T
706
END

B. Description of Subroutine SWITCH(KRATE)

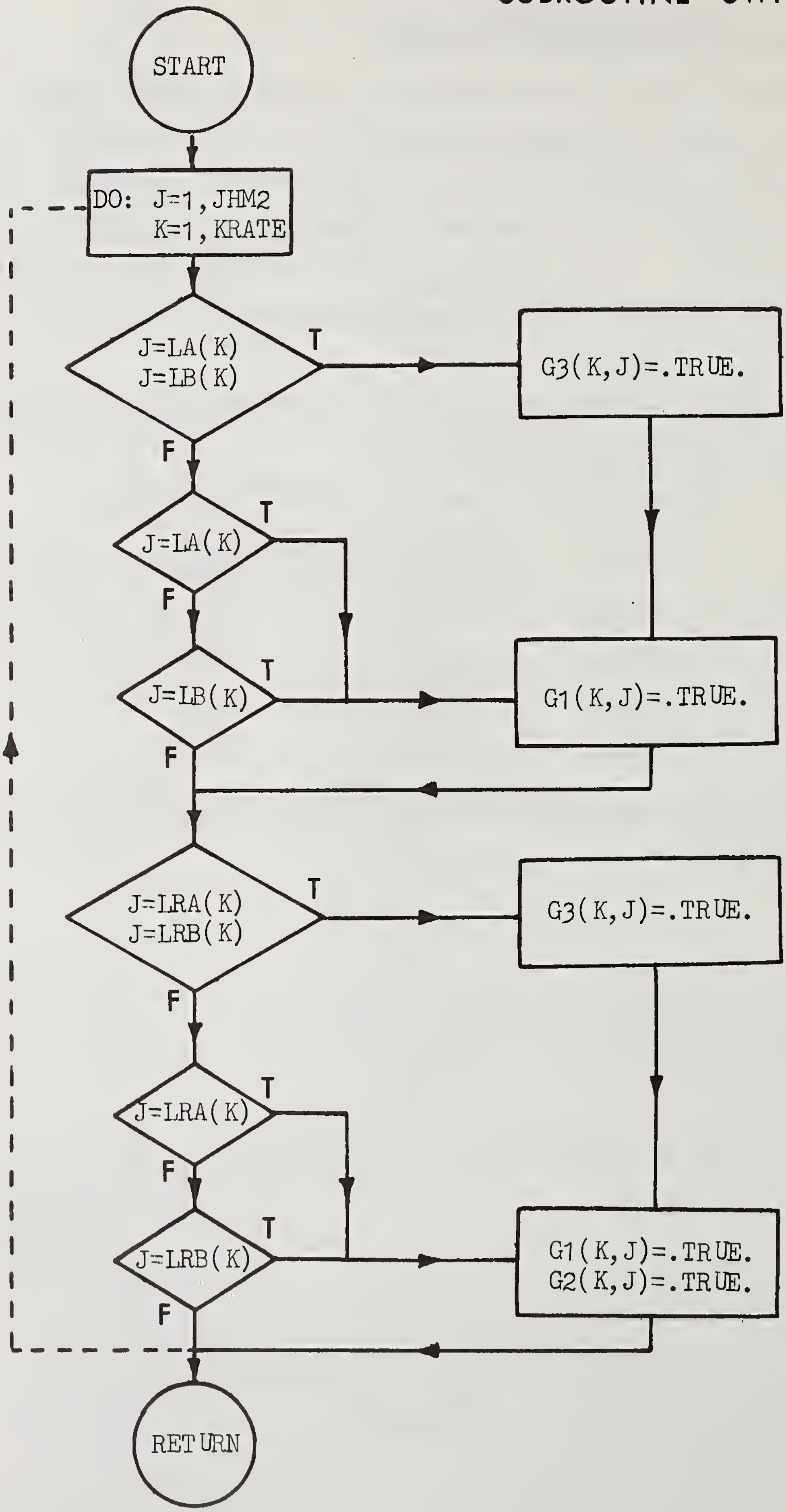
This subroutine sets three flags G1, G2, and G3 which denote the presence and location of a particular species J in a particular reaction K. These flags are initially set to the .FALSE. position. SWITCH sets G3(K,J) to .TRUE. if the species J appears twice on either side of the reaction equation. If it appears on the left side of the equation, G1(K,J) is set to .TRUE.; if on the right side, both G1(K,J) and G2(J,K) are made .TRUE..

As an example, consider the following reactions which are tested for the presence and location of species A.

Reaction	Values of the Flags after Calling SWITCH G1	G2	G3
$A + B \rightleftarrows C + D$	.TRUE.	.FALSE.	.FALSE.
$C + D \rightleftarrows A + B$	.TRUE.	.TRUE.	.FALSE.
$A + A \rightleftarrows C + D$	.TRUE.	.FALSE.	.TRUE.
$C + D \rightleftarrows A + A$	.TRUE.	.TRUE.	.TRUE.
$B + D \rightleftarrows E + F$	.FALSE.	.FALSE.	.FALSE.

These flag values are used in SCHM2 in the calculation of the chemical source terms.

SUBROUTINE SWITCH(KRATE)

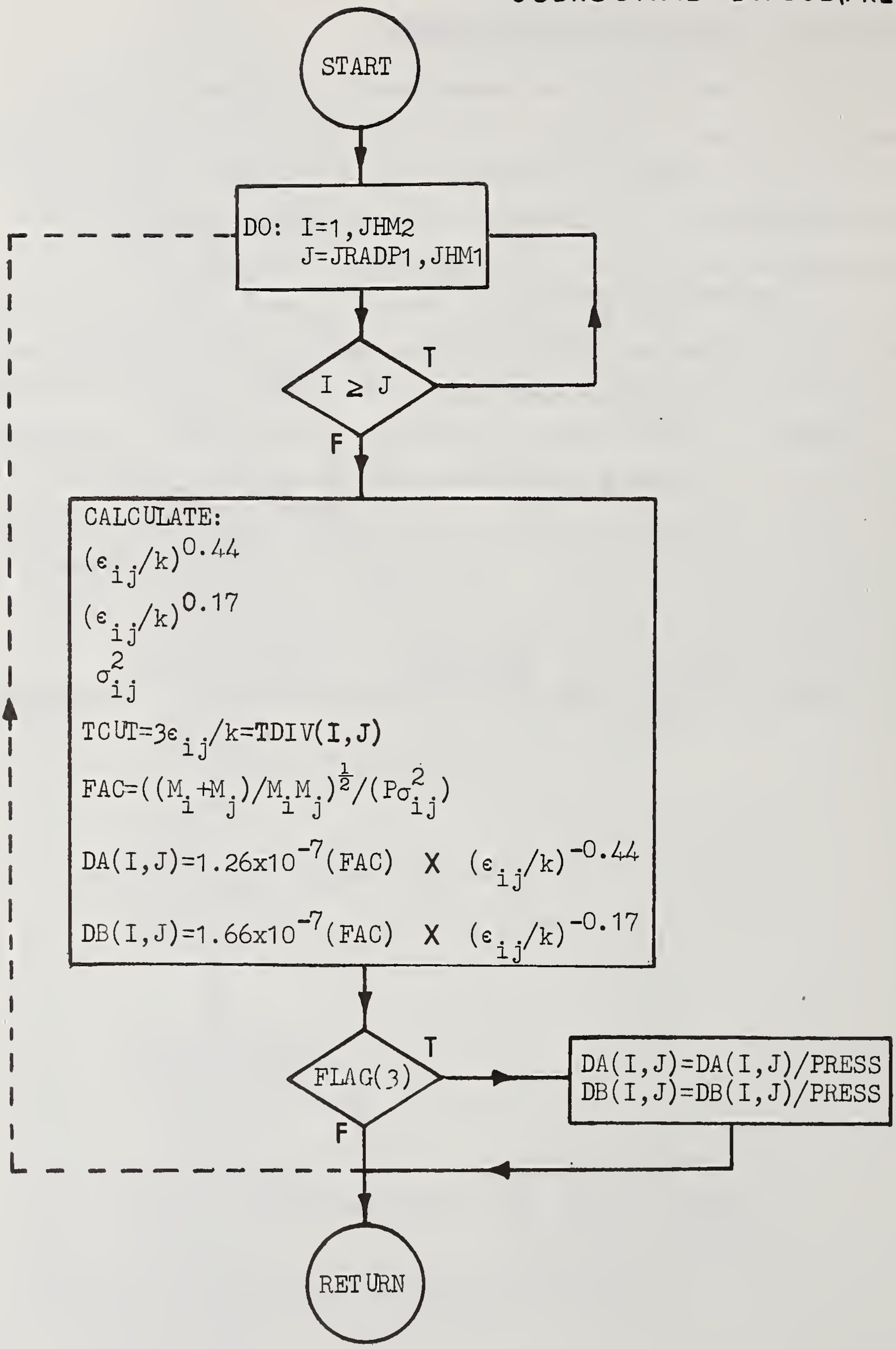


C. Description of Subroutine DIFUSE(PRESS)

This routine calculates the temperature independent portion of the binary diffusion coefficients for the H_2-O_2-N_2 flame from Lennard-Jones parameters. These parameters are assigned values in the DATA statement starting at line DIF00100. M is an array containing the molecular weights in gm/mole; EK is the Lennard-Jones potential parameter ϵ_j/k in ^oK; and S is the parameter σ_j in $\mathring{A}$. These are listed in order of species index. The quantities calculated are DA(I,J) and DB(I,J) which are the temperature independent portion of Eqs. (A72) and (A73), Appendix M, p. A36. Also calculated is the cut-off temperature below which DA(I,J) is used rather than DB(I,J). This is TDIV(I,J) = $3\epsilon_{ij}/k$. I and J are species indices. The index I runs from 1 to JHM2, while J goes from JRADP1 to JHM1. This means that DIFUSE calculates these quantities for interactions between minor and major species, major and major, but not minor and minor.

If FLAG(3) = .TRUE., then DIFUSE divides DA and DB by the pressure PRESS.

SUBROUTINE DIFUSE(PRESS)



D. Description of Subroutine OMEGA

This routine takes the values of N, NII, NEE, and OMR and calculates the grid structure. (See Section III-A, p.11.) The quantities determined are

OM(I) = ω_i

OMD(I) = $\omega_{i+1} - \omega_i$

ROMD(I) = 1./OMD(I)

OMP(I) = $\omega_i + \omega_{i+1}$

BOM(I) = $\omega_{i+1} - \omega_{i-1}$

BOMT3(I) = 3.*BOM(I)

Note that lines OMG00300 and 310 can be removed since the variables OMI and OME which are calculated there are not used.

E. Description of Subroutine INITL/1

This program begins by defining the function $10\omega^3 - 15\omega^4 + 6\omega^5$ which is used to generate S-shaped profiles. It then tests the variable INPRFL(J), where J is the index of a major species. If .FALSE., then the initial profile for J is a decay profile and is calculated from the formula

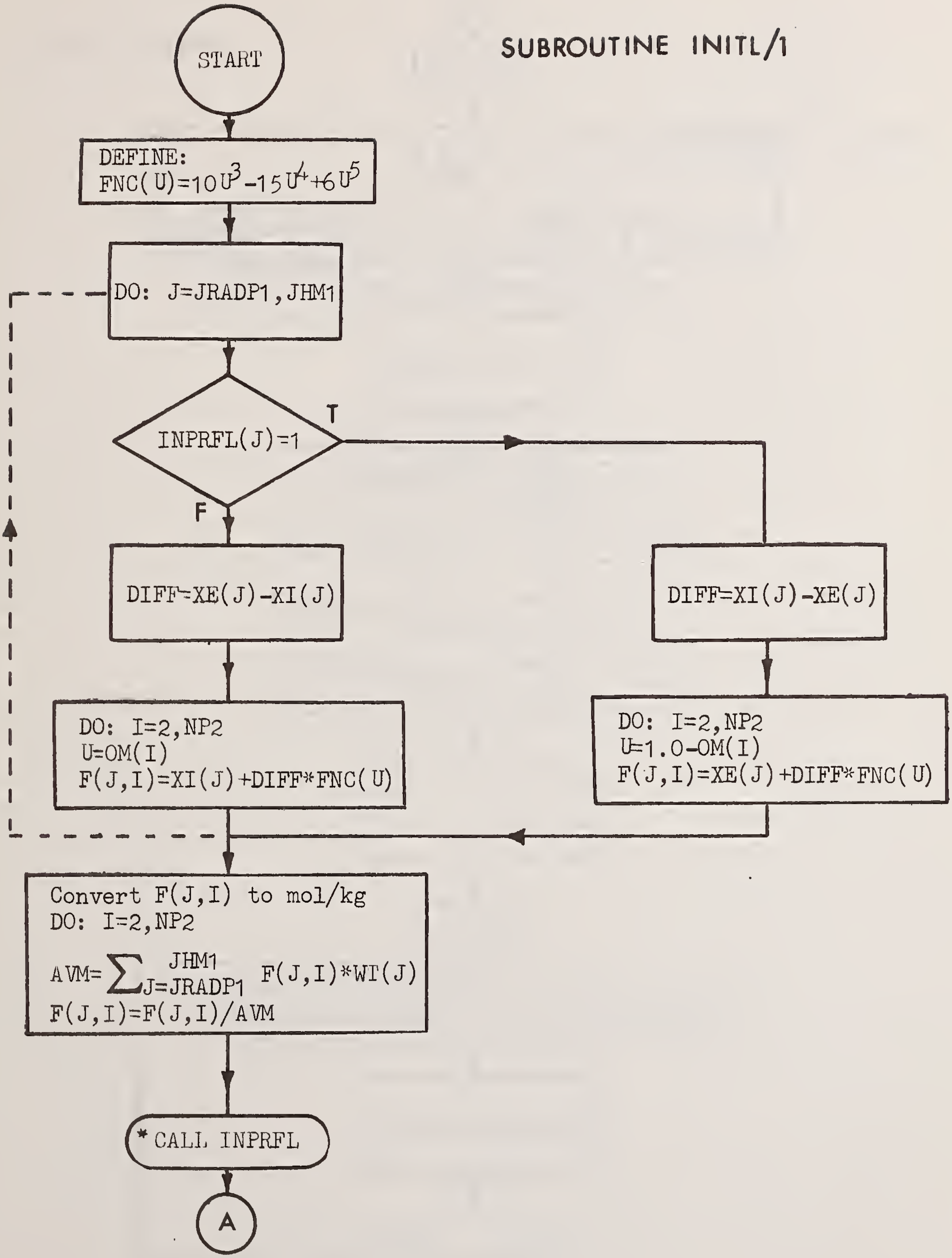
$$XI(J) + (XE(J) - XI(J))(10\omega^3 - 15\omega^4 + 6\omega^5)$$

If .TRUE., then J has a growth profile which is calculated by interchanging XI and XE and inserting $1 - \omega$ for ω in the above formula. This yields mole fraction profiles. To convert them to units of moles/kg, INITL calculates the average molecular weight at a particular grid point from the formula $\langle M \rangle = \sum_j x_j M_j$. The desired profiles are gotten by dividing the mole fraction profiles by $\langle M \rangle$.

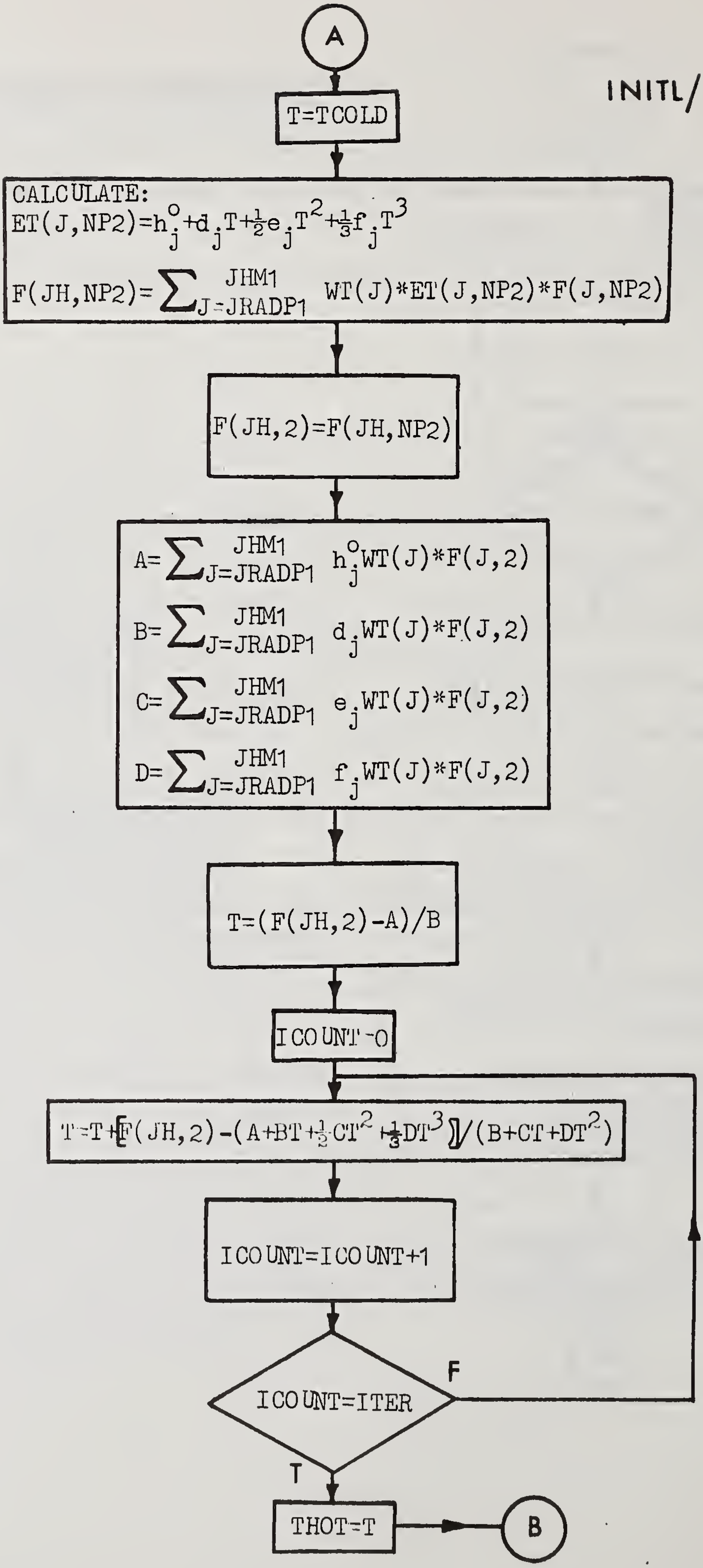
At this point a small subroutine INPRFL is called to set the initial profiles for the trace species. Normally, the trace species are assumed to have ngeligible concentrations at the cold boundary grid points. In setting up INPRFL, do not use equal concentrations at the two boundary grid points, 2 and NP2. This would result in a division by zero on the first call to ENTRN. (See ENTRN, lines ENT00190, 210, and 230.)

INITL next calculates ET(J,JP2), the enthalpy at the cold boundary grid point for all the major species. This will be in units of joules/kg. From these quantities and the composition at the cold boundary, the total enthalpy value h_C = F(JH,NP2) is calculated. The enthalpy at the hot boundary, F(JH,2), is then set equal to h_C and the temperature there is calculated from this and the initial hot side composition by the Newton-Raphson method discussed in Appendix N, p. A39. From the two temperatures TCOLD and THOT an S-shaped temperature profile is calculated. Finally, the enthalpy at each grid point is given the cold boundary grid point value.

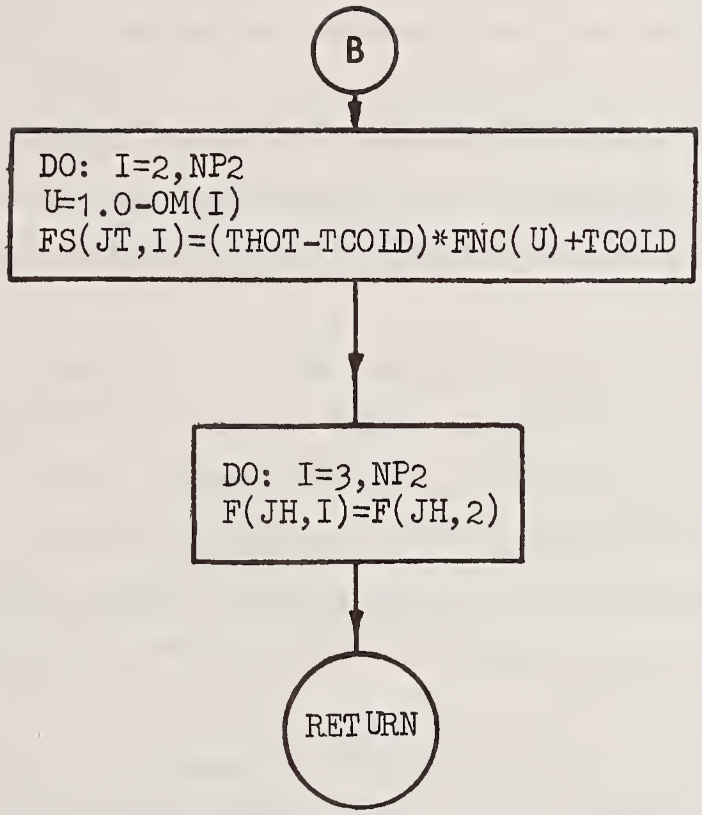
SUBROUTINE INITL/1



INITL/1 (cont.)


A
T=TCOLD
CALCULATE:
$ET(J,NP2)=h^o_j+d_jT+\frac{1}{2}e_jT^2+\frac{1}{3}f_jT^3$
$F(JH,NP2)=\sum_{J=JRADP1}^{JHM1} WT(J)*ET(J,NP2)*F(J,NP2)$
F(JH,2)=F(JH,NP2)
$A=\sum_{J=JRADP1}^{JHM1} h^o_j WT(J)*F(J,2)$
$B=\sum_{J=JRADP1}^{JHM1} d_j WT(J)*F(J,2)$
$C=\sum_{J=JRADP1}^{JHM1} e_j WT(J)*F(J,2)$
$D=\sum_{J=JRADP1}^{JHM1} f_j WT(J)*F(J,2)$
T=(F(JH,2)-A)/B
ICOUNT=0
$T=T+[F(JH,2)-(A+BT+\frac{1}{2}CT^2+\frac{1}{3}DT^3)]/(B+CT+DT^2)$
ICOUNT=ICOUNT+1
ICOUNT=ITER
F
T
THOT=T
B

INITL/1 (cont.)


B
DO: I=2,NP2
U=1.0-OM(I)
FS(JT,I)=(THOT-TCOLD)*FNC(U)+TCOLD
DO: I=3,NP2
F(JH,I)=F(JH,2)
RETURN

F. Description of Subroutine TRANS(ICALC)/6

This subroutine calculates the temperature, the mean molecular weight, the density, and the transport parameters for three flames; ozone decomposition, H_2-Br_2, and H_2-O_2-N_2. It begins by calculating the temperature and its reciprocal from the enthalpy by the Newton-Raphson procedure described in Appendix N, p. A39. At the same time it also calculates the mean molecular weight and its reciprocal at each grid point.

The quantity ICALC is then tested; if it equals zero, then TRANS jumps to the next grid point. Under this condition the transport properties are not calculated. If ICALC equals one, then TRANS continues by calculating the density RHO(I) at each grid point. FLAG(3) is tested, and if .TRUE., the density is multiplied by PRESS which, in this case, will be different from unity; if .FALSE., then this multiplication is not made.

FLAG(7) is tested and if .TRUE., transport properties are calculated for the ozone decomposition flame. The desired parameters are $\Delta_j\rho^2$ for diffusion, and $\lambda\rho/C_p$ for thermal conductivity. The average heat capacity C_p (=CPMIX) is first calculated and from this and the density, $\lambda\rho/C_p$ (=PREF(JH,I)) is calculated from the expression $2.2026\text{x}10^{-3}T^{\frac{1}{2}}\rho/C_p$. For this flame a Lewis number of 0.94 was used so that the quantities $\Delta_j\rho^2$ (=PREF(J,I)) are all given the values 0.94*PREF(JH,I). The program then jumps to statement 57.

If FLAG(7) was .FALSE., FLAG(8) is tested; if .TRUE., transport properties are calculated for the H_2-Br_2 flame. We used the following expressions to calculate the diffusion parameters;

$$\Delta_{Br}\rho^2 = 0.155\text{x}10^{-4}(T/TCOLD)^{1.67}\rho^2$$

$$\Delta_H\rho^2 = 1.05\text{x}10^{-4}(T/TCOLD)^{1.67}\rho^2$$

$$\Delta_{H_2}\rho^2,\ \Delta_{Br_2}\rho^2,\ \Delta_{HBr}\rho^2 = 0.101\text{x}10^{-4}(T/TCOLD)^{1.67}\rho^2$$

The thermal conductivity of the mixture was taken to be

$\lambda\rho/C_p = 3.34944\text{x}10^{-2}(T/TCOLD)^{0.67}\rho/CPMIX.$

The program then jumps to statement 57.

If FLAG(8) was .FALSE., FLAG(1) is tested; if .TRUE., the program jumps to the next grid point. This is the situation where we are using constant values for the transport parameters. If FLAG(1) = .FALSE., then TRANS proceeds to calculate the transport properties for the H_2-O_2-N_2 flame. It begins by calculating the binary diffusion coefficients using the values of DA(I,J) and DB(I,J) determined in DIFUSE. Since each species at a particular grid point uses either $T^{1.94}$ or $T^{1.67}$ as the factor multiplying DA or DB, it is desirable to avoid doing these exponentiations more than once per grid point. This feat is accomplished by assigning .TRUE. values to the variables TAGA and TAGB (initially .FALSE.) if either $T^{1.94}$ or $T^{1.67}$ has already been calculated for a particular species. Testing these flags on subsequent passes through the DO loop allows one to avoid repeated evaluation of $T^{1.94}$ or $T^{1.67}$. The temperature dependent binary diffusion coefficients are stored in the array DD(KI,KJ), where KI and KJ are species indices.

The diffusion coefficients Δ_j are then calculated from the DD(KI,KJ) values and the concentrations F(J,I) by means of Eq. (A9), Appendix C, p. A7. Diffusion parameters PREF(J,I) = $\Delta_j\rho^2$ are then calculated.

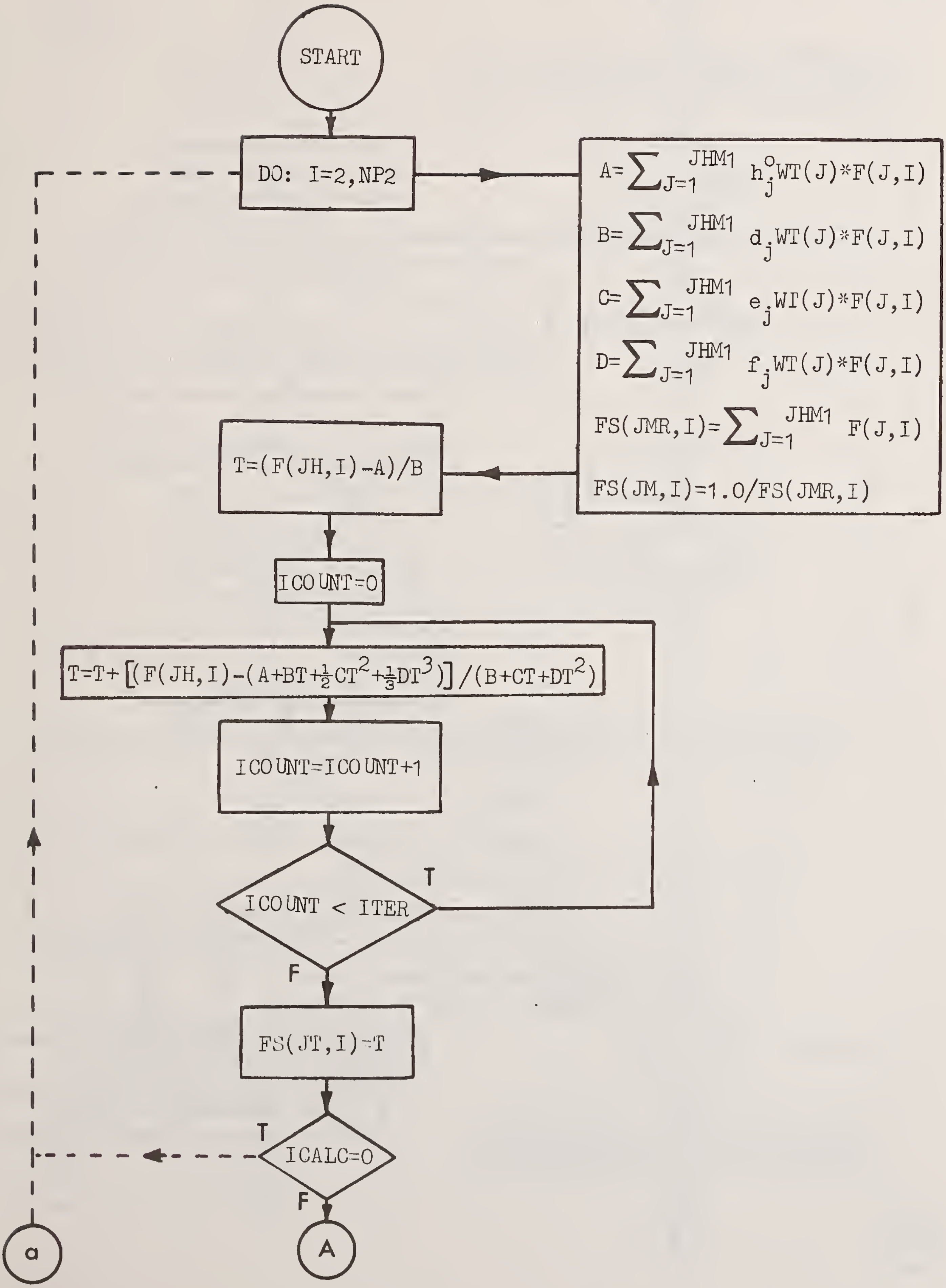
TRANS next calculates the thermal conductivity of the mixture from the formula given in Appendix D, p. A8. For this, it is necessary to have values of the temperature dependent coefficients A_{ij} defined by Eq. (A11), Appendix D, p. A8. These were calculated from this formula by a <u>separate</u> program, and the results fit by a least squares procedure to a power series in the temperature. These expressions for the A_{ij} are given by lines TRN01510 through 01620. Note that the values are stored in the DD array since the binary diffusion coefficients previously stored there are no longer needed. Also given just before these are the power series expansions of the thermal conductivities of the pure major species; lines TRN01420 through 01450. These are placed in CON(J). Note that it is for these arrays that the indices LO2, LH2O, LH2, and LN2 are required. From the A_{ij} and CON(J) values, the mixture thermal conductivity is calculated. This is then converted to the thermal conductivity parameter PREF(JH,I) = $\lambda\rho/C_p$.

FLAG(5) is then tested. If .FALSE., TRANS goes to the next grid point. If .TRUE., the enthalpy source term is being calculated and values of ENT(J,I), the enthalpy of each species at each grid point are determined.

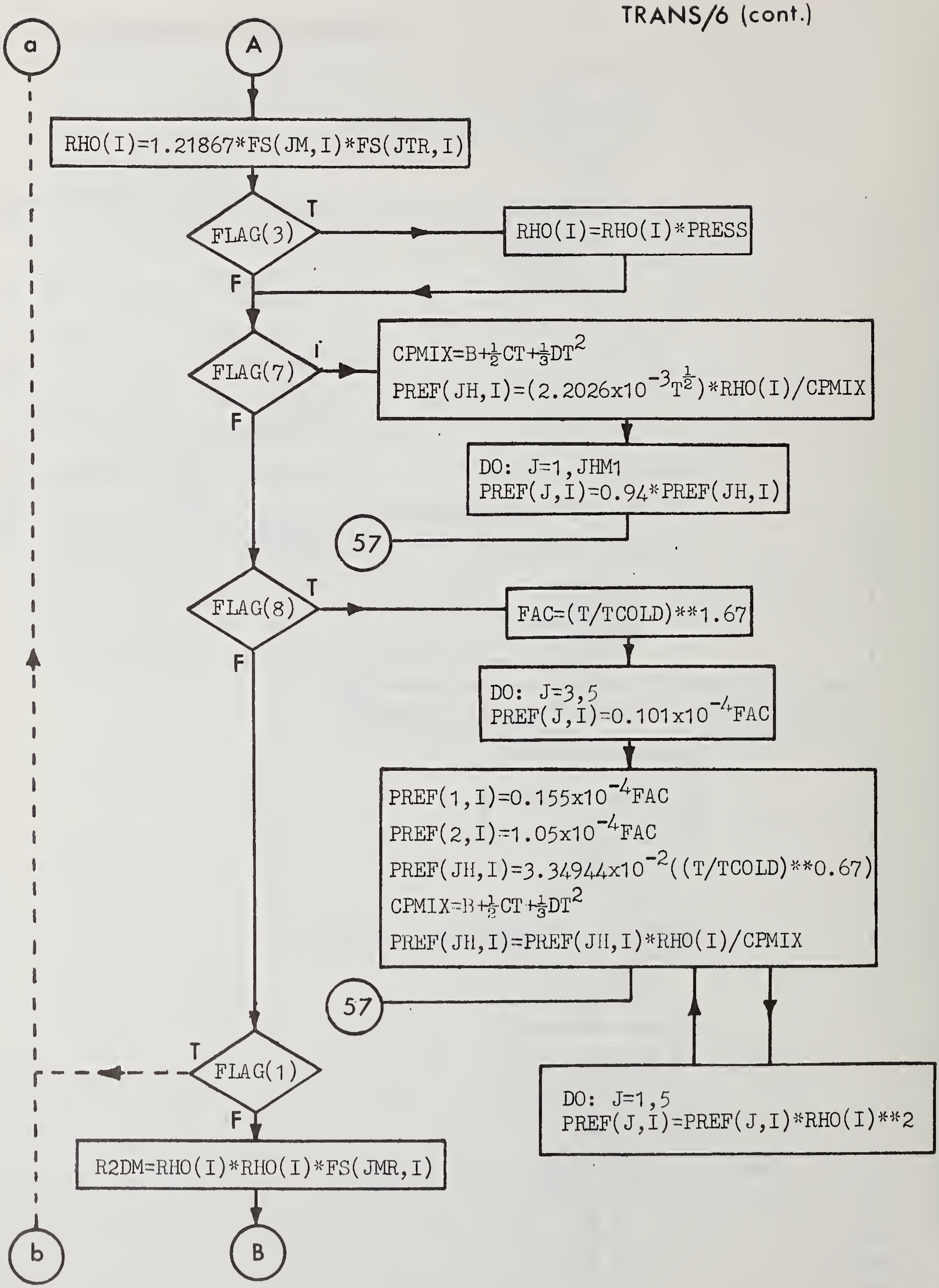
After the calculations have been completed for all the grid points, ICALC is tested. If equal to zero, a return to SPALD occurs. If equal to one, TRANS proceeds to calculate the values of the reciprocal of the mean molecular weight at the control volume boundaries. This is taken to be the average value of FS(JMR,I) at two adjacent grid points, and is stored in RMB(I). The same thing is also done for the diffusion and thermal conductivity parameters, and the results placed in PREF(J,I) and PREF(JH,I). Thus, these arrays will contain the control volume boundary values and not the grid point values.

If FLAG(5) = .TRUE., several quantities appearing in the enthalpy source term are evaluated. Note that both parts of the enthalpy source term are calculated together and not separately as in the case of the chemical source term.

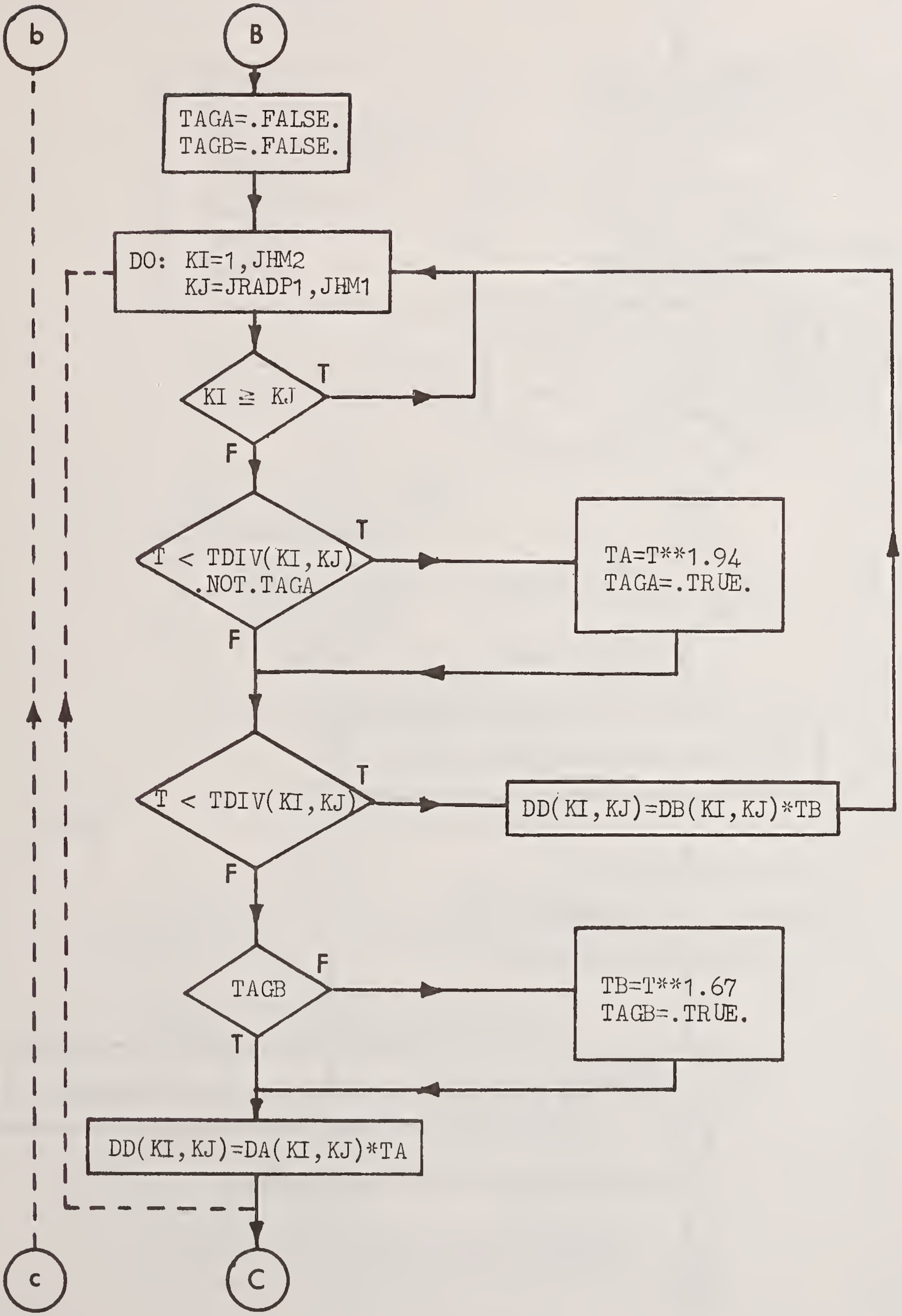
SUBROUTINE TRANS(ICALC) / 6



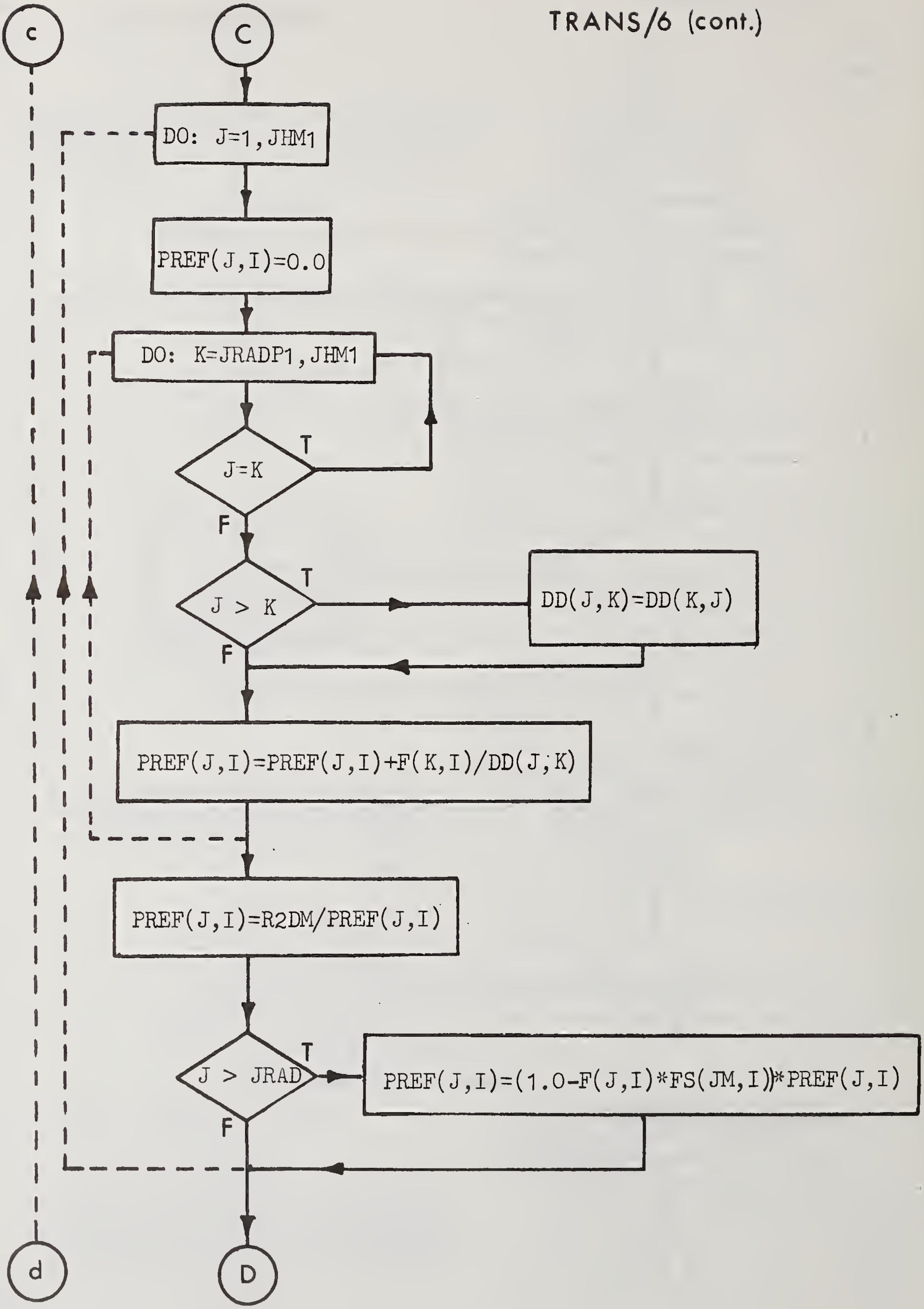
TRANS/6 (cont.)


a
A
RHO(I)=1.21867*FS(JM,I)*FS(JTR,I)
FLAG(3)
T
F
RHO(I)=RHO(I)*PRESS
FLAG(7)
F
CPMIX=B+½CT+⅓DT²
PREF(JH,I)=(2.2026x10⁻³T½)*RHO(I)/CPMIX
DO: J=1,JHM1
PREF(J,I)=0.94*PREF(JH,I)
57
FLAG(8)
T
F
FAC=(T/TCOLD)**1.67
DO: J=3,5
PREF(J,I)=0.101x10⁻⁴FAC
PREF(1,I)=0.155x10⁻⁴FAC
PREF(2,I)=1.05x10⁻⁴FAC
PREF(JH,I)=3.34944x10⁻²((T/TCOLD)**0.67)
CPMIX=B+½CT+⅓DT²
PREF(JH,I)=PREF(JH,I)*RHO(I)/CPMIX
57
DO: J=1,5
PREF(J,I)=PREF(J,I)*RHO(I)**2
FLAG(1)
T
F
R2DM=RHO(I)*RHO(I)*FS(JMR,I)
b
B

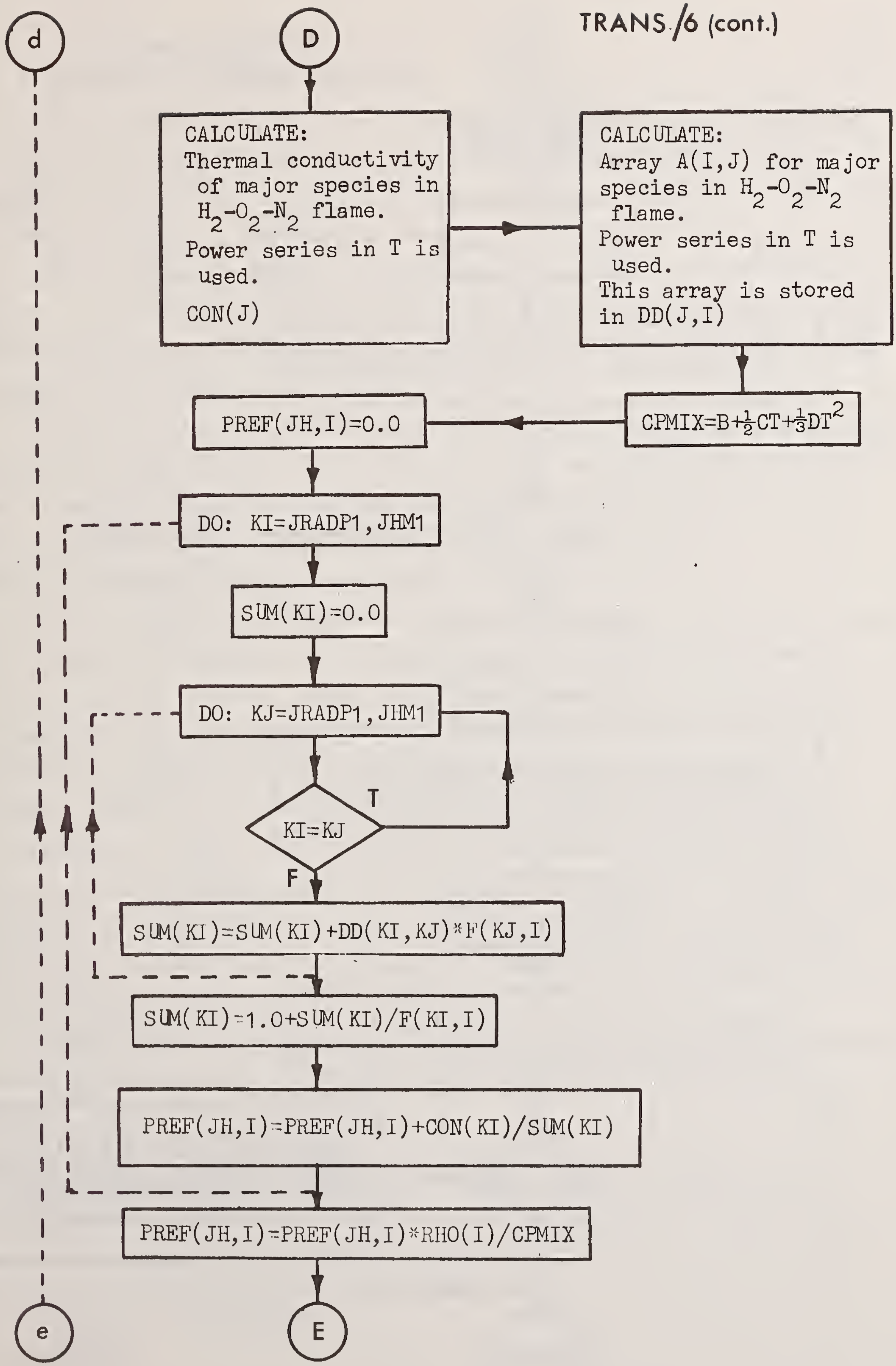
TRANS/6 (cont.)


b
B
TAGA=.FALSE.
TAGB=.FALSE.
DO: KI=1,JHM2
KJ=JRADP1,JHM1
KI ≧ KJ
T
F
T < TDIV(KI,KJ)
.NOT.TAGA
T
TA=T**1.94
TAGA=.TRUE.
F
T < TDIV(KI,KJ)
T
DD(KI,KJ)=DB(KI,KJ)*TB
F
TAGB
F
TB=T**1.67
TAGB=.TRUE.
T
DD(KI,KJ)=DA(KI,KJ)*TA
c
C

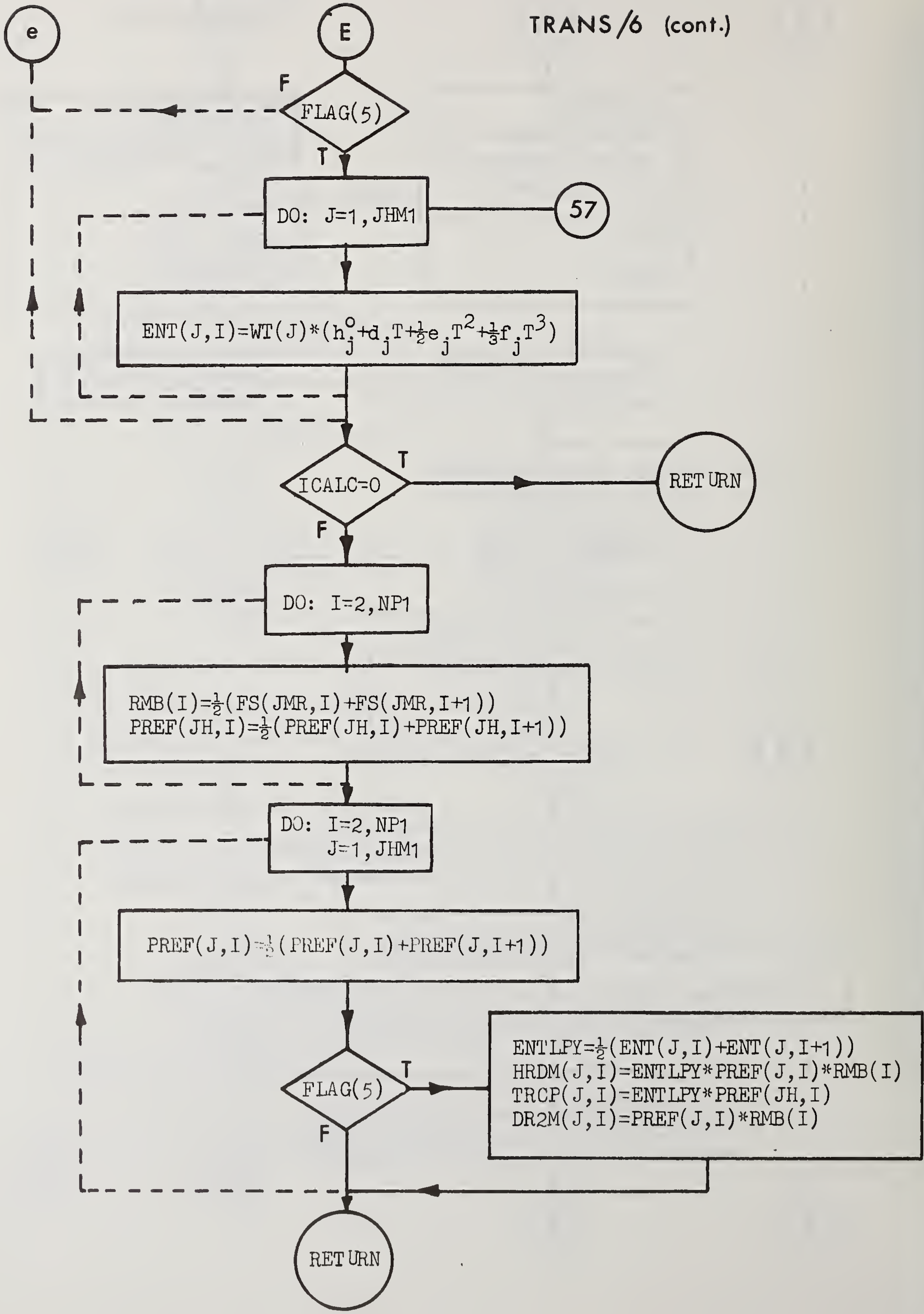
TRANS/6 (cont.)


c
C
DO: J=1,JHM1
PREF(J,I)=0.0
DO: K=JRADP1,JHM1
J=K
T
F
J > K
T
DD(J,K)=DD(K,J)
F
PREF(J,I)=PREF(J,I)+F(K,I)/DD(J,K)
PREF(J,I)=R2DM/PREF(J,I)
J > JRAD
T
PREF(J,I)=(1.0-F(J,I)*FS(JM,I))*PREF(J,I)
F
d
D

TRANS./6 (cont.)


d
D
CALCULATE:
Thermal conductivity of major species in H2-O2-N2 flame.
Power series in T is used.
CON(J)
CALCULATE:
Array A(I,J) for major species in H2-O2-N2 flame.
Power series in T is used.
This array is stored in DD(J,I)
CPMIX=B+½CT+⅓DT²
PREF(JH,I)=0.0
DO: KI=JRADP1,JHM1
SUM(KI)=0.0
DO: KJ=JRADP1,JHM1
KI=KJ
T
F
SUM(KI)=SUM(KI)+DD(KI,KJ)*F(KJ,I)
SUM(KI)=1.0+SUM(KI)/F(KI,I)
PREF(JH,I)=PREF(JH,I)+CON(KI)/SUM(KI)
PREF(JH,I)=PREF(JH,I)*RHO(I)/CPMIX
e
E

TRANS/6 (cont.)



e
E
F
FLAG(5)
T
DO: J=1,JHM1
57
ENT(J,I)=WT(J)*(h_j^o+d_jT+½$e_j$$T^2$+⅓$f_j$$T^3$)
ICALC=0
T
RETURN
F
DO: I=2,NP1
RMB(I)=½(FS(JMR,I)+FS(JMR,I+1))
PREF(JH,I)=½(PREF(JH,I)+PREF(JH,I+1))
DO: I=2,NP1
J=1,JHM1
PREF(J,I)=½(PREF(J,I)+PREF(J,I+1))
FLAG(5)
T
ENTLPY=½(ENT(J,I)+ENT(J,I+1))
HRDM(J,I)=ENTLPY*PREF(J,I)*RMB(I)
TRCP(J,I)=ENTLPY*PREF(JH,I)
DR2M(J,I)=PREF(J,I)*RMB(I)
F
RETURN

G. Description of Subroutine RATCN/3

This subroutine takes the Arrhenius parameters and calculates the forward rate constants for each reaction; from these, and the free energies of reaction, it then calculates the rate constants for the reverse reactions.

RATCN starts by calculating the quantities $(RT)^{-1}$ at each grid point. R is the gas constant and has the value 8.2057×10^{-5} atm-m^3/mole-deg. These are stored in RGT(I) and are used later in the calculation of the reverse rate constants.

The calculation of the forward rate constants starts by the testing of FFLAG(K); if .FALSE., RATCN sets RATE(K,I) to zero and goes to the next K value. This option allows one to examine the effect of neglecting the forward rate without having to change the Arrhenius parameters which may have been used in a previous calculation.

Next, BFLAG(K) and EFLAG(K) are tested together; if both are .FALSE., then both β and E/R are zero for this reaction and so exponentiation can be avoided. In this case RATE(K,I) is given the value FRQ(K), the Arrhenius frequency factor, and RATCN moves to the next K. If either BFLAG(K) or EFLAG(K) or both are .TRUE., BFLAG(K) is tested and if .FALSE., RATE(K,I) is given the value FRQ(K)*DEXP(-EACT(K)/FS(JT,I)). If BFLAG(K) = .TRUE., then EFLAG(K) is tested and if .FALSE., RATE(K,I) takes the value FRQ(K)*(FS(JT,I)**BETA(K)). If EFLAG(K) = .TRUE., then RATE(K,I) is given the value FRQ(K)*FS(JT,I)**BETA(K))*DEXP(-EACT(K)/FS(JT,I)). Only in this last case is it necessary to calculate the complete Arrhenius expression.

After calculating all of the forward rate constants, RATCN evaluates all of the equilibrium constants. Consider the general reaction,

$$R_1 + R_2 + \dots\dots\dots\dots + R_r \underset{k'}{\overset{k}{\rightleftarrows}} P_1 + P_2 + \dots\dots\dots\dots + P_p$$

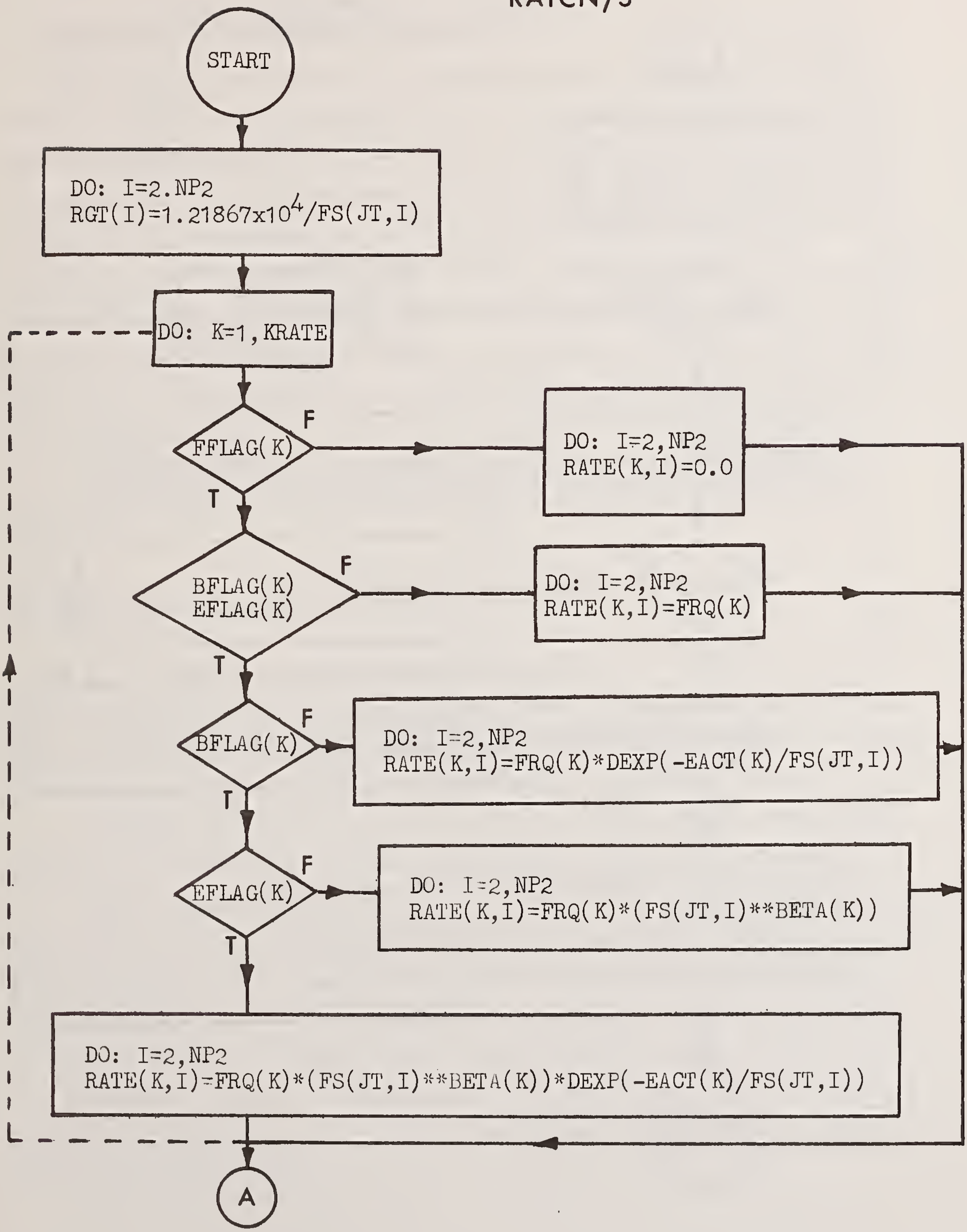
R_i and P_i represent one molecule of a species, so two symbols may represent the same molecule. The equilibrium constant in terms of the partial pressures of the various species is

$$K_p = \frac{\Pi_{i=1}^{p} P_{P_i}}{\Pi_{i=1}^{r} P_{R_i}} = \exp(-\Delta F^o/RT)$$

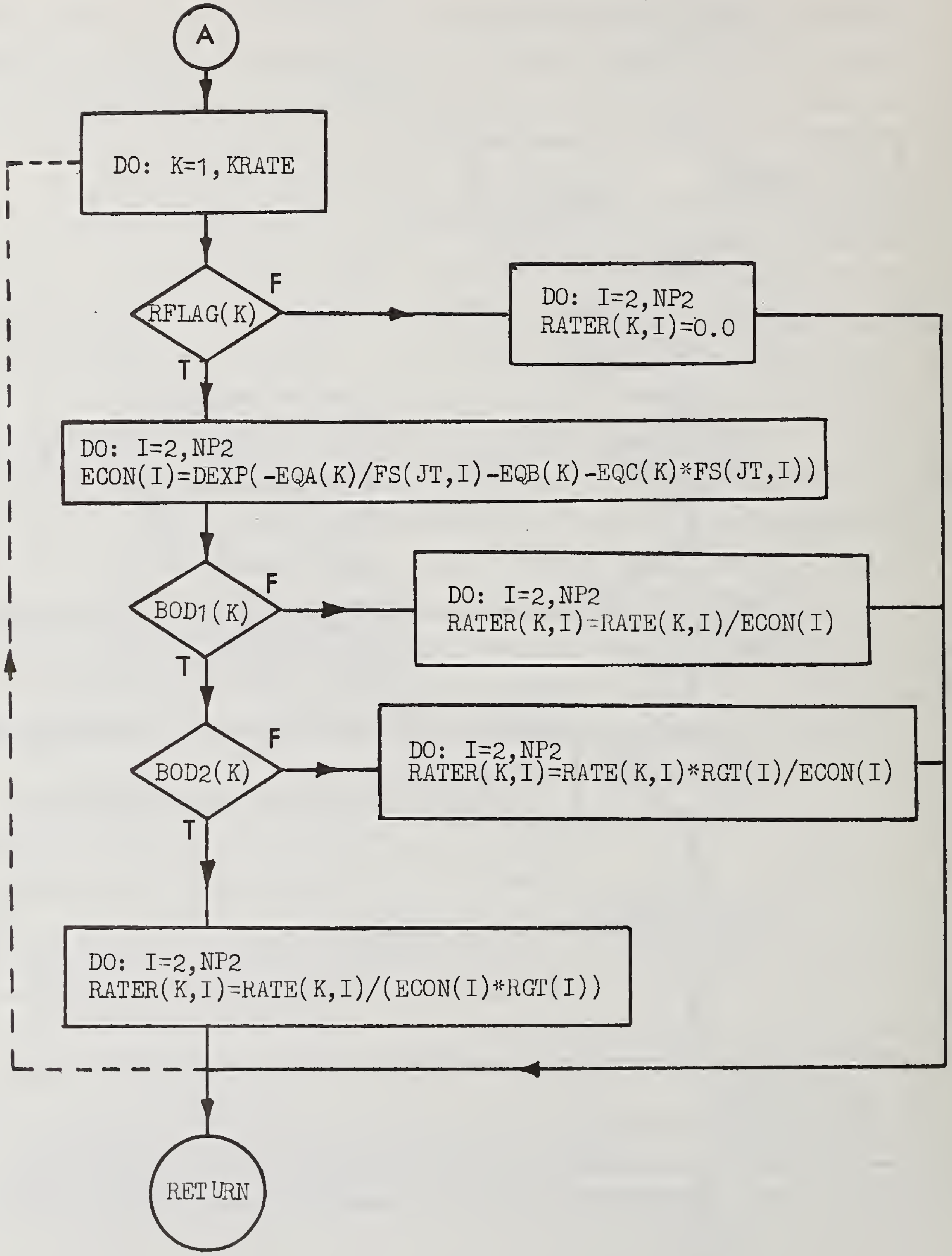
We want the equilibrium constant in terms of concentrations; this is related to K_p by the expression $K_c = K_p(RT)^{-n}$, where n = p - r and R is the gas constant in atm-m^3/mole-deg. Since $K_c = k/k'$, we have for the reverse rate constant $k' = k(RT)^n/K_p$.

To calculate k' RATCN firsts tests RFLAG(K); if .FALSE., it sets the reverse rate constant k' (=RATER(K,I)) to zero and goes to the next K value. If this flag is .TRUE., then it calculates $-\Delta F^o/RT$ from the expression -EQA(K)/T - EQB(K) - EQC(K)*T, evaluates exp($-\Delta F^o/RT$), and storesthe results in ECON(I). (Note that EQA(K), EQB(K), and EQC(K) as read by SPALD give ΔF^o in kcal/mole. SPALD immediately multiplies them by 1000/R, where R = 1.9869 cal/mole-deg to give $\Delta F^o/R$.) BOD1(K) is tested next; if .FALSE., then the number of molecules does not change in the reaction; n = 0 and RATER(K,I) is given the value RATE(K,I)/ ECON(I). If BOD1(K) is .TRUE., then we have a 3-body reaction with a unity change in the number of molecules. BOD2(K) is tested to find out the sign of n. If BOD2(K) = .FALSE., then n = -1 and RATER(K,I) is given the value RATE(K,I)*RGT(I)/ECON(I); if .TRUE., then n = +1 and it has the value RATE(K,I)/(ECON(I)*RGT(I)). (Remember that RGT(I) = $(RT)^{-1}$.)

SUBROUTINE
RATCN/3



RATCN/3 (cont.)


A
DO: K=1,KRATE
RFLAG(K)
F
T
DO: I=2,NP2
RATER(K,I)=0.0
DO: I=2,NP2
ECON(I)=DEXP(-EQA(K)/FS(JT,I)-EQB(K)-EQC(K)*FS(JT,I))
BOD1(K)
F
T
DO: I=2,NP2
RATER(K,I)=RATE(K,I)/ECON(I)
BOD2(K)
F
T
DO: I=2,NP2
RATER(K,I)=RATE(K,I)*RGT(I)/ECON(I)
DO: I=2,NP2
RATER(K,I)=RATE(K,I)/(ECON(I)*RGT(I))
RETURN

H. Description of Subroutine ENTRN(J,IFOW)

The purpose of ENTRN is to calculate the so-called entrainment rates $\dot{m}_C$ (=RME) and $\dot{m}_H$ (=RMI). RME is the mass flow rate in kg/m^2-s across the cold boundary and RMI that across the hot boundary. When a steady-state is attained, RME and RMI become equal. Their values depend on the concentrations of the species JENTRN at the hot and cold boundaries and at the points NEE and NII. The grid width PEI is changed if RME and RMI have different values. By this means the grid can be kept centered about the region containing the largest concentration gradients. (See Appendix J, p. A26.) This routine also calculates the flame velocity.

During the course of the integration ENTRN is called with J = JENTRN and IFOW = 0. The routine begins by evaluating the integral

$$\eta \int_0^1 \frac{R_j}{\rho M_j} d\omega,$$

where R_j is the mass production rate of species j in kg/m^3-s. The quantity $R_j/\rho M_j$ has units of moles/kg-s and is the quantity stored in the array R(J,I). An approximation with three grid points is used for this integral. This is given by Eq. (A61), Appendix J, p. A27.

The flame velocity is gotten by dividing the value of the above integral by the density at the cold boundary grid point, RHO(NP2).

IFOW is then tested; if equal to zero, ENTRN continues by calculating RME and RMI from Eqs. (A58), (A59), and (A60), Appendix J, p. A26. The grid width PEI is then incremented by the amount (RMI - RME)*DX.

The condition IFOW = 1 is used at an output point. Then ENTRN is called for J = 1 through JHM1 to calculate the flame velocity from the production rates of each species. In principle, these velocities should be the same in the steady-state flame. When IFOW = 1, RMI and RME are not calculated.

I. Description of the Subroutine SCHM1/1

This short routine calculates the chemical source term arising from the diffusion of mean molecular weight. (See Eq. (A57), Appendix I, p. A24.) Concentration values at the control volume boundaries are required; these are calculated from the average of two adjacent grid points and stored in BF(J,I). The other quantities required are all evaluated elsewhere. Values of this source term are placed in the array SC1(J,I).

J. Description of Subroutine SCHM2(J)/7

This subroutine calculates the part of the chemical source term arising from the chemical reactions.

SCHM2 begins by testing G1(K,J); if .FALSE., then species J does not appear in reaction K and the program moves to the next K value. If .TRUE., then K is tested to see if it equals 5 or 10. These are the reactions $H + OH + M \rightleftarrows H_2O + M$ and $H + O_2 + M \rightleftarrows HO_2 + M$ in the H_2-O_2-N_2 flame which have the special third-body concentrations FS(JBODYA,I) and FS(JBODYB,I). This routine can also be used for the ozone decomposition and the H_2-Br_2 flames because both require fewer than 5 reactions. To use SCHM2 for other flames, it will be necessary to remove the following lines:

SC200260

SC200270

SC200420

⋮

SC200480

If K is not equal to 5 or 10, then BOD1(K) is tested; if .FALSE., K does not contain a three-body reaction and SCHM2 proceeds to calculate the forward and reverse rates of the reaction. These are put into FOR(I) and REV(I). At this point, these rates are in units of moles-m^3/kg^2-s and will later be multiplied by the density to give them in units of moles/kg-s. If BOD1(K) = .TRUE., then K contains a three-body reaction and BOD2(K) is tested to find out whether it is the forward or the reverse reaction. If BOD2(K) = .FALSE., then it is the forward reaction and FOR(I) and REV(I) are calculated accordingly. If BOD2(K) = .TRUE., then the opposite situation holds. FOR(I) and REV(I) are now combined in various ways depending on the values of the flags G1, G2, and G3. The results are placed in SU(J,I) and SD(J,I). This process is repeated for all the K values, and the results added to SU(J,I) and SD(J,I).

After multiplying SU(J,I) and SD(J,I) by the density RHO(I), the production rate of species J, R(J,I), is obtained by adding them together. The final desired quantities are gotten by dividing the value in

SD(J,I) by the concentration of J, F(J,I), and multiplying both SD(J,I) and SU(J,I) by $\frac{1}{2}\eta\Omega$.

For an example of how this routine functions consider the reaction mechanism discussed in Appendix I, p. A21. There are two reactions

$$A + B \underset{k_1'}{\overset{k_1}{\rightleftarrows}} C + D$$

$$A + A + M \underset{k_2'}{\overset{k_2}{\rightleftarrows}} E + M$$

We want to calculate the source term parameters $S_P(A)$ and $S_F(A)$ for the species A defined in Eq. (A53), p. A23.

```
For K = 1, we have BOD1(1) = .FALSE.
                   BOD2(1) = .FALSE.
                   G1(1,LA) = .TRUE.
                   G2(1,LA) = .FALSE.
                   G3(1,LA) = .FALSE.
For K = 2, we have BOD1(2) = .TRUE.
                   BOD2(2) = .FALSE.
                   G1(2,LA) = .TRUE.
                   G2(2,LA) = .FALSE.
                   G3(2,LA) = .TRUE.
```

When K = 1, SCHM2 places in SU(LA,I) and SD(LA,I), respectively, the quantities $k_1'\varphi_C\varphi_D$ and $- k_1\varphi_A\varphi_B$. Moving to K = 2, it <u>adds</u> to SU(LA,I) and SD(LA,I), the quantities $2(k_2\varphi_A\varphi_A\varphi_M\rho + k_2'\varphi_E\varphi_M)$ and $- 4k_2\varphi_A\varphi_A\varphi_M\rho$. Note that φ_M, the concentration of the buffer species, is taken to be the reciprocal of the mean molecular weight, FS(JMR,I). SU(LA,I) and SD(LA,I) are then both multiplied by the density. At this point they have the values

$$SU(LA,I) = k_1'\rho\varphi_C\varphi_D + 2k_2\rho^2\varphi_A\varphi_A\varphi_M + 2k_2'\rho\varphi_E\varphi_M$$

$$SD(LA,I) = - k_1\rho\varphi_A\varphi_B - 4k_2\rho^2\varphi_A\varphi_A\varphi_M$$

The production rate R(LA,I) of species A is the sum of SU(LA,I) and SD(LA,I).

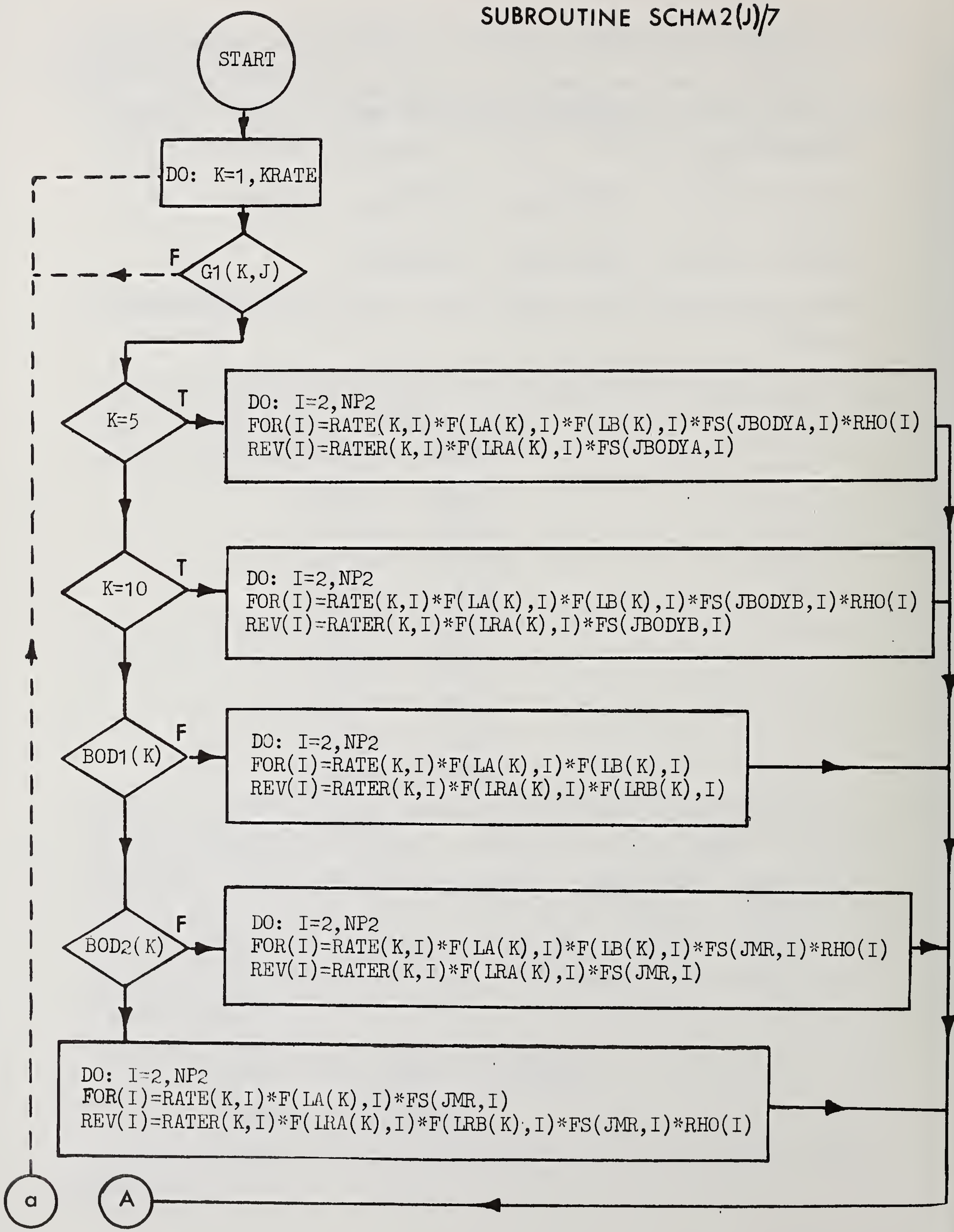
SU(LA,I) and SD(LA,I) are not yet equal to the parameters $S_P(A)$ and $S_F(A)$. These are obtained by dividing SD(LA,I) by φ_A and multiplying both SD(LA,I) and SU(LA,I) by $\frac{1}{2}\eta\Omega = \frac{1}{2}\eta(\omega_{i+1} - \omega_{i-1})$ = PEIOM2(I). The result is

$$SU(LA,I) = S_P(A) = \tfrac{1}{2}\eta\Omega(k_1'\rho\varphi_C\varphi_D + 2k_2\rho^2\varphi_A^2\varphi_M + 2k_2'\rho\varphi_E\varphi_M)$$

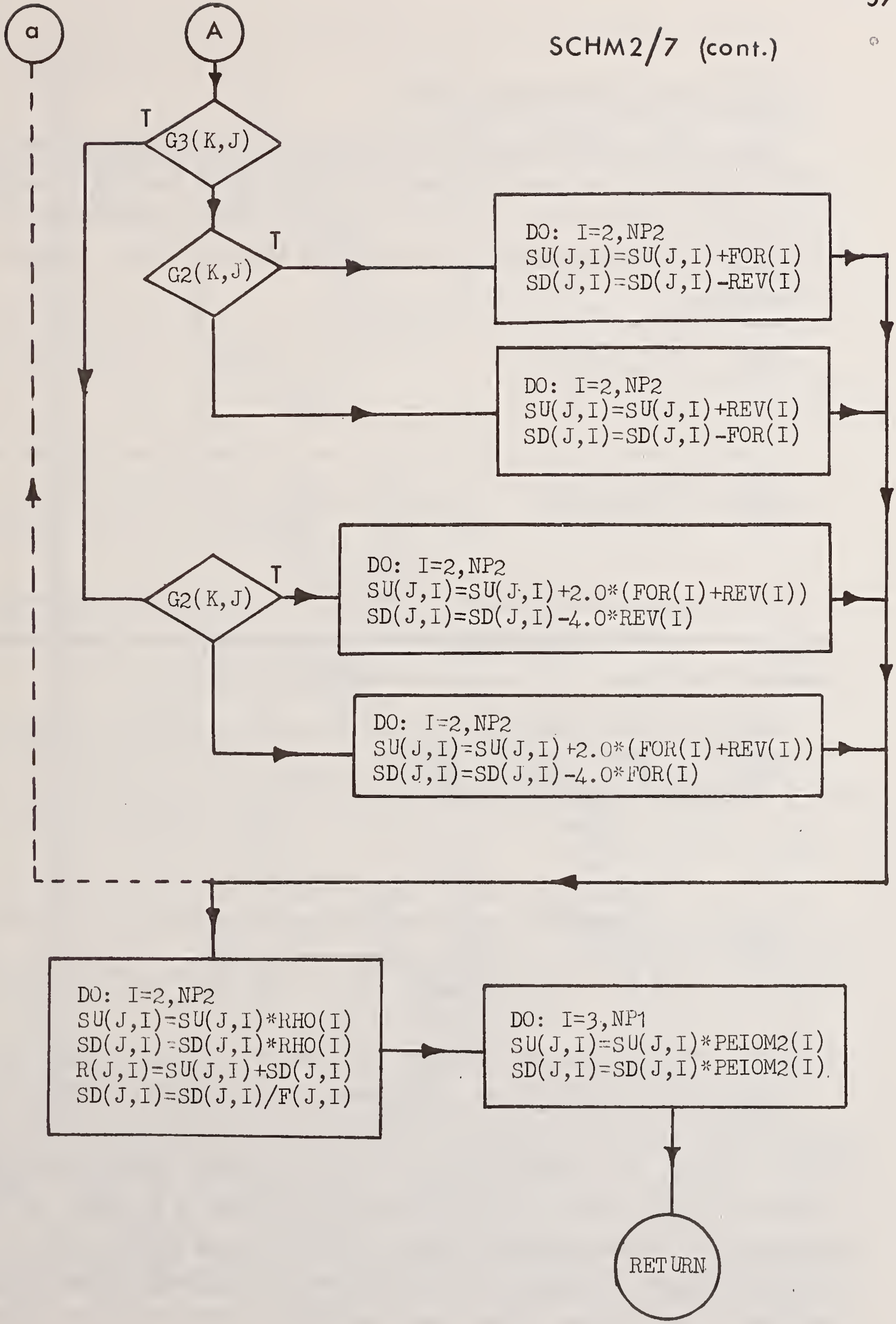
$$SD(LA,I) = S_F(A) = \tfrac{1}{2}\eta\Omega(-k_1\rho\varphi_B - 4k_2\rho^2\varphi_A\varphi_M)$$

This is the same as that shown in Eqs. (A51) and (A52), Appendix I, p. A23.

SUBROUTINE SCHM2(J)/7



SCHM2/7 (cont.)


a
A
G3(K,J)
T
G2(K,J)
T
DO: I=2,NP2
SU(J,I)=SU(J,I)+FOR(I)
SD(J,I)=SD(J,I)-REV(I)
DO: I=2,NP2
SU(J,I)=SU(J,I)+REV(I)
SD(J,I)=SD(J,I)-FOR(I)
G2(K,J)
T
DO: I=2,NP2
SU(J,I)=SU(J,I)+2.0*(FOR(I)+REV(I))
SD(J,I)=SD(J,I)-4.0*REV(I)
DO: I=2,NP2
SU(J,I)=SU(J,I)+2.0*(FOR(I)+REV(I))
SD(J,I)=SD(J,I)-4.0*FOR(I)
DO: I=2,NP2
SU(J,I)=SU(J,I)*RHO(I)
SD(J,I)=SD(J,I)*RHO(I)
R(J,I)=SU(J,I)+SD(J,I)
SD(J,I)=SD(J,I)/F(J,I)
DO: I=3,NP1
SU(J,I)=SU(J,I)*PEIOM2(I)
SD(J,I)=SD(J,I)*PEIOM2(I)
RETURN

K. Description of Subroutine CALC(J)

This is the subroutine which solves the finite-difference equations (A42), Appendix H, p. A19. To see how this program functions, consider a 4 point grid system (N = 3), with the grid index running from 2 to 5. There will be two unknowns, Φ_3 and Φ_4, for which Eq. (A42) yields two equations,

$$\Phi_3 = \frac{A'_3}{D_3}\Phi_4 + \frac{B'_3}{D_3}\Phi_2 + \frac{C'_3}{D_3}$$

$$\Phi_4 = \frac{A'_4}{D_4}\Phi_5 + \frac{B'_4}{D_4}\Phi_3 + \frac{C'_4}{D_4}$$

A', B', C', and D are given by Eqs. (A40), p. A18 and the values Φ_2 and Φ_5 at the hot and cold boundaries, respectively, are known from the boundary conditions. By simple elimination, we obtain

$$\Phi_3 = \frac{\dfrac{A'_3A'_4}{D_3D_4}\Phi_5 + \dfrac{A'_3C'_4}{D_3D_4} + \dfrac{B'_3}{D_3}\Phi_2 + \dfrac{C'_3}{D_3}}{1 - \dfrac{A'_3B'_4}{D_3D_4}} \tag{15}$$

$$\Phi_4 = \frac{A'_4}{D_4}\Phi_5 + \frac{B'_4}{D_4}\left\{\frac{\dfrac{A'_3A'_4}{D_3D_4}\Phi_5 + \dfrac{A'_3C'_4}{D_3D_4} + \dfrac{B'_3}{D_3}\Phi_2 + \dfrac{C'_3}{D_3}}{1 - \dfrac{A'_3B'_4}{D_3D_4}}\right\} + \frac{C'_4}{D_4} \tag{16}$$

Let us go through CALC step-by-step and see how it solves these equations.

There are 5 arrays to be considered; A(J,I), B(J,I), C(J,I), D(J,I), F(J,I), and also a variable T. Initially, A(J,I), B(J,I), C(J,I), and D(J,I) contain the values of A', B', C', and D which were evaluated by SPALD with Eqs. (A40). F(J,I) contains the values of Φ_i from the previous step. The contents of A(J,I), B(J,I), T, and F(J,I) as CALC is executed change as follows: (Note that NP1 = 4, and the number in parentheses is the number of the step executed.)

A(J,3)	(0) A'_3	(1) $\frac{A'_3}{D_3}$
A(J,4)	(0) A'_4	(4) $\frac{A'_4}{D_4 - \frac{B'_4 A'_3}{D_3}}$
B(J,3)	(0) B'_3	(2) $\frac{B'_3 \Phi_2 + C'_3}{D_3}$
B(J,4)	(0) B'_4	(5) $\frac{B'_4\left(\frac{B'_3\Phi_2}{D_3} + \frac{C'_3}{D_3}\right) + C_4}{D_4 - \frac{B'_4 A'_3}{D_3}}$
T	(0) 0.0	(3) $D_4 - \frac{B'_4 A'_3}{D_3}$
F(J,3)	(0) Φ_3(old)	(7) $\frac{A'_3}{D_3}\left\{\frac{A'_3\Phi_5 + B'_4\left(\frac{B'_3\Phi_2 + C'_3}{D_3}\right) + C'_4}{D_4 - \frac{B'_4 A'_3}{D_3}}\right\} + \frac{B'_3\Phi_2 + C'_3}{D_3}$
F(J,4)	(0) Φ_4(old)	(6) $\frac{A'_4\Phi_5 + B'_4\left(\frac{B'_3\Phi_2 + C'_3}{D_3}\right) + C'_4}{D_4 - \frac{B'_4 A'_3}{D_3}}$

A little algebra will show that the contents of F(J,3) and F(J,4) after these 7 steps is the same as that given by Eqs. (15) and (16).

After solving (A42) for all the grid points, CALC tests for negative species concentrations. If the test is positive, it puts a very small positive number in F(J,I).

L. Description of Subroutine SENTP/1

This routine calculates the total enthalpy source term which is given by the sum of Eqs. (A55) and (A56), Appendix I, p. A24. To see how it works consider the enthalpy equation (A23), Appendix F, p. A11. The complete source term is

$$\mathcal{S} = \frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left\{\sum_j h_j^*\left(\frac{\Delta_j\rho^2}{\langle M\rangle}\frac{\partial(\varphi_j\langle M\rangle)}{\partial\omega} - \frac{\lambda\rho}{C_p}\frac{\partial\varphi_j}{\partial\omega}\right)\right\}$$

SENTP calculates

$$S = \eta\int_-^+ \mathcal{S}\,d\omega = \frac{1}{\eta}\sum_j\left[\left\{\frac{h_j^*\Delta_j\rho^2}{\langle M\rangle}\right\}_+\left\{\frac{\partial(\varphi_j\langle M\rangle)}{\partial\omega}\right\}_+ - \left\{\frac{h_j^*\Delta_j\rho^2}{\langle M\rangle}\right\}_-\left\{\frac{\partial(\varphi_j\langle M\rangle)}{\partial\omega}\right\}_-\right.$$

$$\left. -\left\{\frac{h_j^*\lambda\rho}{C_p}\right\}_+\left\{\frac{\partial\varphi_j}{\partial\omega}\right\}_+ + \left\{\frac{h_j^*\lambda\rho}{C_p}\right\}_-\left\{\frac{\partial\varphi_j}{\partial\omega}\right\}_-\right]$$

The transport parameters were calculated in TRANS and represent values at the control volume boundaries. They are

$$\text{HDRM(J,I)} = \left\{\frac{h_j^*\Delta_j\rho^2}{\langle M\rangle}\right\}_+$$

$$\text{HDRM(J,I-1)} = \left\{\frac{h_j^*\Delta_j\rho^2}{\langle M\rangle}\right\}_-$$

$$\text{TRCP(J,I)} = \left\{\frac{h_j^*\lambda\rho}{C_p}\right\}_+$$

$$\text{TRCP(J,I-1)} = \left\{\frac{h_j^*\lambda\rho}{C_p}\right\}_-$$

The derivatives are approximated by

$$\left\{\frac{\partial(\varphi_j\langle M\rangle)}{\partial\omega}\right\}_+ \approx \text{(F(J,I+1)*FS(JM,I+1) - F(J,I)*FS(JM,I))*ROMD(I)}$$

etc.,

where ROMD(I) = $1/(\omega_{i+1} - \omega_i)$

It would be desirable to rewrite SENTP so that the part arising from the diffusion of the mean molecular weight was calculated separately. Thus, it could easily be neglected if desired.

M. Description of Subroutine ZCALC/1

This short routine calculates the value of the laboratory spatial variable y_i (=Y(I)) which corresponds to a particular value of the spatial variable ω_i in the Spalding coordinate system. It does this by evaluating the integral

$$y = \eta \int_0^{\omega} \frac{d\omega}{\rho}$$

by means of the approximation given by Eq. (A70), Appendix L, p. A34.

N. Description of Subroutine OUTPUT(L,HEADNG,RUNID,X,INPUT,LMAX)

This subroutine generates and controls the printing of a variety of output information. At intervals of LPRINT steps during the integration a single page of output is generated. This contains the values at each grid point of the species concentrations, the enthalpy, and the temperature. It also prints the velocities calculated from the production rates of each species, the grid width PEI, and the entrainment rates RME and RMI. When L = LMAX, the integration is stopped and 11 pages of additional information are printed:

Page 1) Values at each grid point of ω, the laboratory spatial variable y in meters, the density in kg/m^3, the mean molecular weight $\langle M \rangle$ in kg/mole, and the total heat release rate in J/kg-s.

2) The production rates R(J,I) in moles/kg-s for each species at each grid point.

3) The transport parameters at each grid point; i.e., the array PREF(J,I), $\Delta_j \rho^2$ and $\lambda\rho/C_p$ in kg^2/m^4-s.

4,5) Values of each rate constant, both forward and reverse rates, at each grid point.

6,7) The rates, forward and reverse, of each reaction at each grid point in moles/kg-s.

8) The maximum rates, forward and reverse, of each reaction and the grid point index at which they occur.

9) A list of the maximum rates of each reaction ranked in decending values. For each maximum rate, its K value and direction are specified.

10,11) Heat release rates, both forward and reverse, of each reaction at each grid point in J/kg-s.

OUTPUT begins by printing the step number L, the step size DX, and the integration time X. It then prints the heading for the profile colums and follows with F(J,I), F(JH,I), FS(JT,I), VEL(J), PEI, RMI, and RME.

It tests for L = LMAX; if .TRUE., it proceeds by printing the values of the grid parameters, N, NII, NEE, and OMR; the index parameters JH, JRAD, JM, JMR, JT, JTR, JBODYA, JBODYB; the temperature on the cold side of the flame TCOLD, the pressure PRESS, and the value of INPUT.

FLAG(2) is tested; if .FALSE., then the buffer concentration is being kept constant. This means that its production rate R(JHM1,I) is zero. In this case, the production rate for the next major species R(JHM2,I) was not calculated by SCHM2. OUTPUT calculates it from the relation $\sum_J^{JHM1} R(J,I)WT(J) = 0.0$. This section is skipped if FLAG(2) = .TRUE..

OUTPUT next calculates the quantities PUT(J) = $h_j M_j = h_j^*$, and FOR(I) = $\sum_J^{JHM1} R(J,I)*PUT(J)$. This latter quantity is the total heat release rate in J/kg-s. It then prints OM(I), Y(I), RHO(I), FS(JM,I), and FOR(I), followed by the arrays R(J,I) and PREF(J,I).

It then tests FLAG(9); if .TRUE., it prints the rate constants. This requires two pages. Each column represents a grid point whose index value appears as the heading. There are 16 columns per page giving a total of 32 grid positions. This may not be sufficient to cover all the grid points. Since points near the boundaries are not very interesting, we start printing at I = ISTART, where ISTART = 4 in the present version of OUTPUT. Its value can, of course, be changed to suit the user's requirements. The next page starts printing at ISTART = ISTART + 16 which equals 20 in the present case.

FLAG(10) is then tested; if .TRUE., the reactions rates are calculated at each point in the flame. Note that FOR(I) which contained the total heat release rate is now used for the forward reaction rate. The rates are calculated from scratch; i.e., from the values of the rate constants RATE(K,I), RATER(K,I) and the concentrations F(J,I). Also calculated at this time are the heat release rates for the forward and reverse reactions. These are stored in the arrays HV(K,I), and HW(K,I), respectively. Note that we have treated the cases K = 5 and 10 in the same manner as discussed in SCHM2. To use OUTPUT for flames (other then H_2-O_2-N_2) which have more than 4 reactions, it is necessary

to remove the following lines:

OUT01680

OUT01690

OUT01960

⋮

OUT 02100

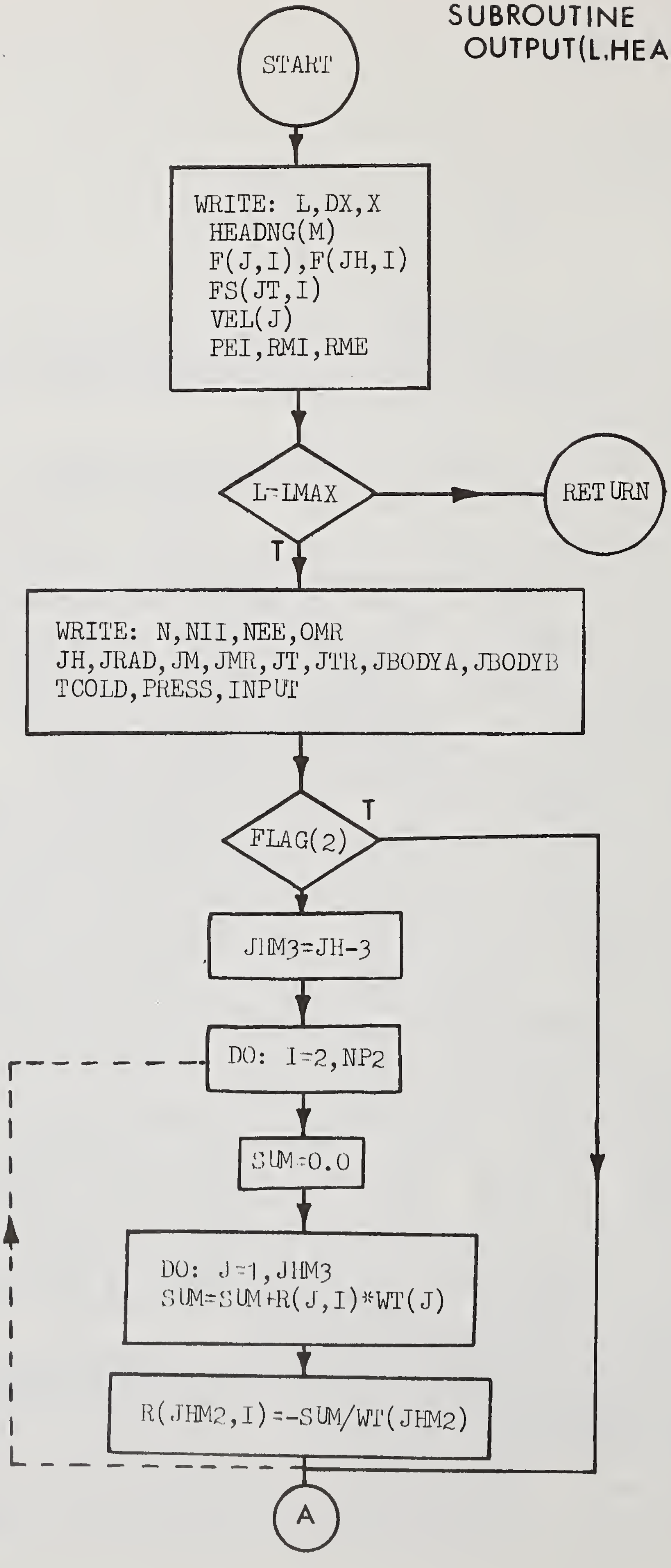
The reaction rates are then printed on two pages in the same format as the rate constants were printed. Note that ISTART also appears in this portion of the routine.

The maximum rate of each reaction is now determined. This is accomplished by running through the grid index I for each reaction K and setting the variable XMAX(K) equal to V(K,I) if V(K,I) is greater than the previous value of XMAX(K). V(K,I) is the rate of the forward reaction and is in single precision. The same thing is done for the reverse reaction and yields YMAX(K). XMAX(K) and YMAX(K) are then printed along with the grid index value to which they correspond.

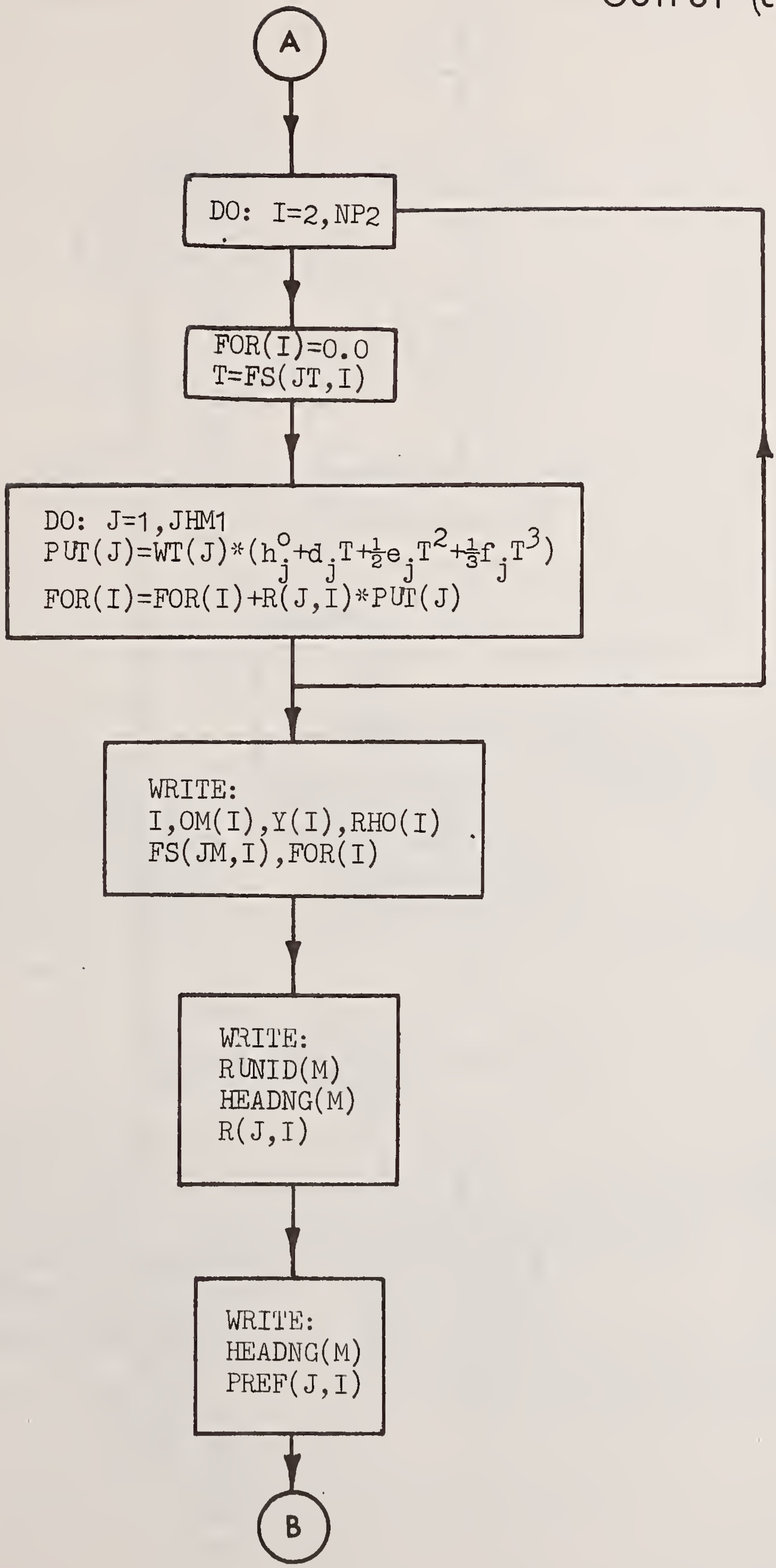
A new single precision variable Z(K) is then defined; Z(K) = XMAX(K) and Z(K+KRATE) = YMAX(K) for K = 1, KRATE. Thus Z(K) runs from K = 1 to 2*KRATE. The ranking of the maximum values of the rates is accomplished by running through Z(K) and setting ZTAB = Z(K) if Z(K) is greater then the previous value of ZTAB. KTAB is set equal to the value of K for this maximum - maximum rate. Also defined is a logical variable SKIP(K), K = 1, 2*KRATE, set initially to the .FALSE. position. At this point OUTPUT sets SKIP(KTAB) = .TRUE.. Then the Z(K) are re-exmined for the maximum value, but now Z(KTAB) is omitted. This is accomplished by testing SKIP(K); if .TRUE., then KTAB is skipped and the next K value is compared with ZTAB. The list from which the maximum value is selected thus becomes smaller as each maximum is removed. In this way the ranking is accomplished. The maximum values of each reaction are then printed in order of decreasing magnitude.

Finally, OUTPUT prints two pages of heat release rates for each reaction. Note that ISTART appears here also.

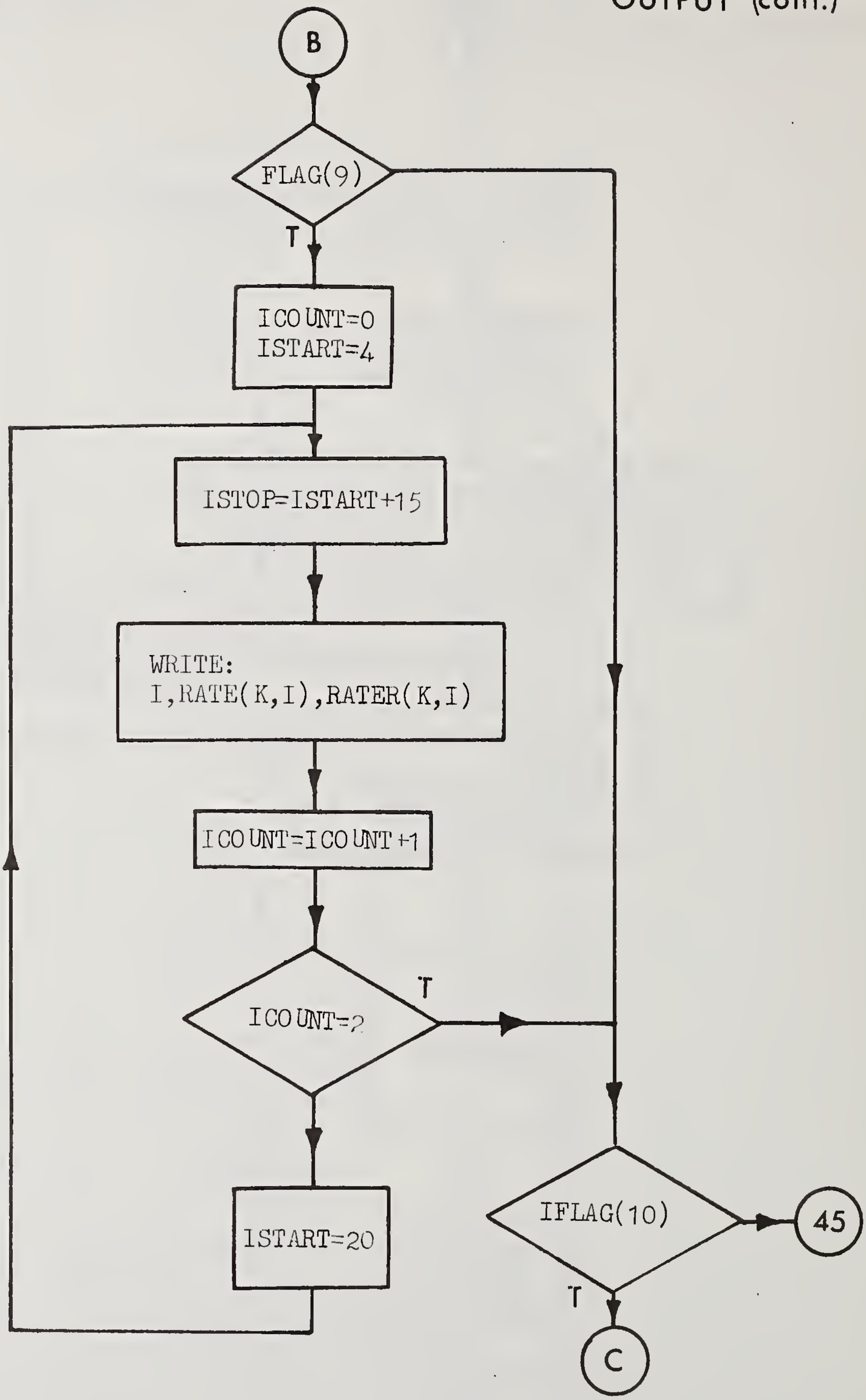
SUBROUTINE
OUTPUT(L,HEADNG,RUNID,X,INPUT,LMAX)



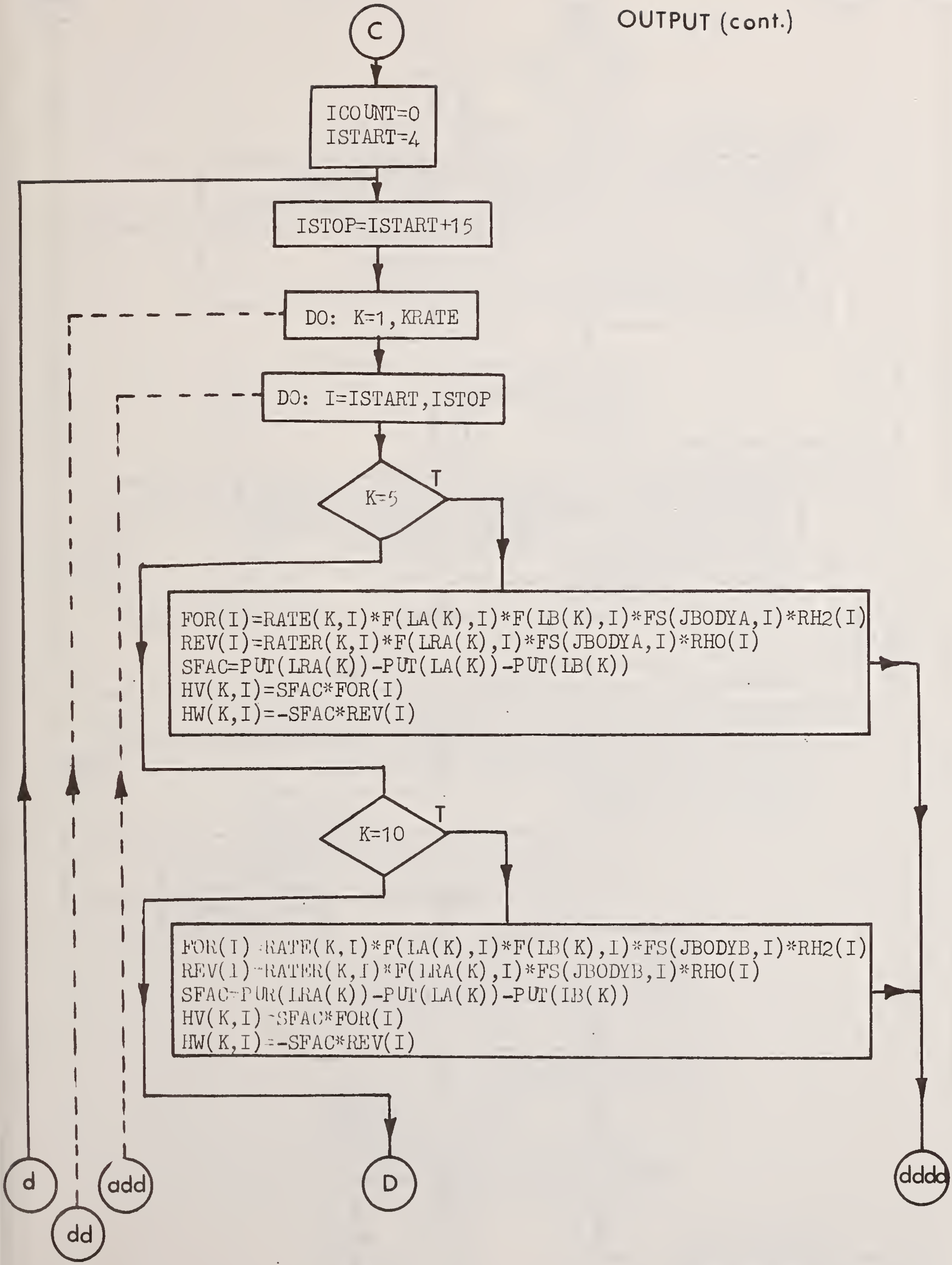
OUTPUT (cont.)


A
DO: I=2,NP2
FOR(I)=0.0
T=FS(JT,I)
DO: J=1,JHM1
PUT(J)=WT(J)*($h_j^o+d_jT+\frac{1}{2}e_jT^2+\frac{1}{3}f_jT^3$)
FOR(I)=FOR(I)+R(J,I)*PUT(J)
WRITE:
I,OM(I),Y(I),RHO(I)
FS(JM,I),FOR(I)
WRITE:
RUNID(M)
HEADNG(M)
R(J,I)
WRITE:
HEADNG(M)
PREF(J,I)
B

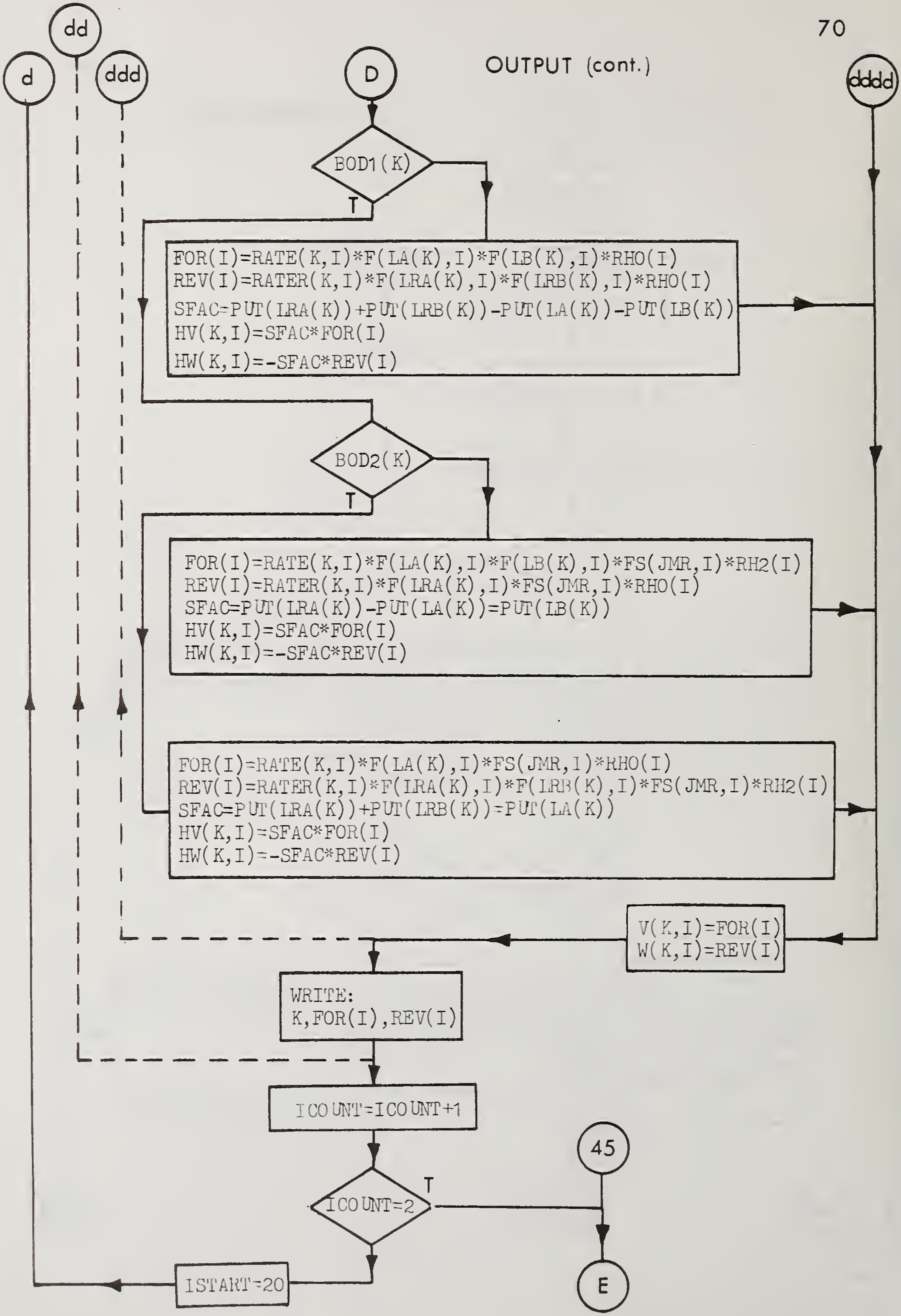
OUTPUT (cont.)


B
FLAG(9)
T
ICOUNT=0
ISTART=4
ISTOP=ISTART+15
WRITE:
I,RATE(K,I),RATER(K,I)
ICOUNT=ICOUNT+1
ICOUNT=2
T
ISTART=20
IFLAG(10)
45
T
C

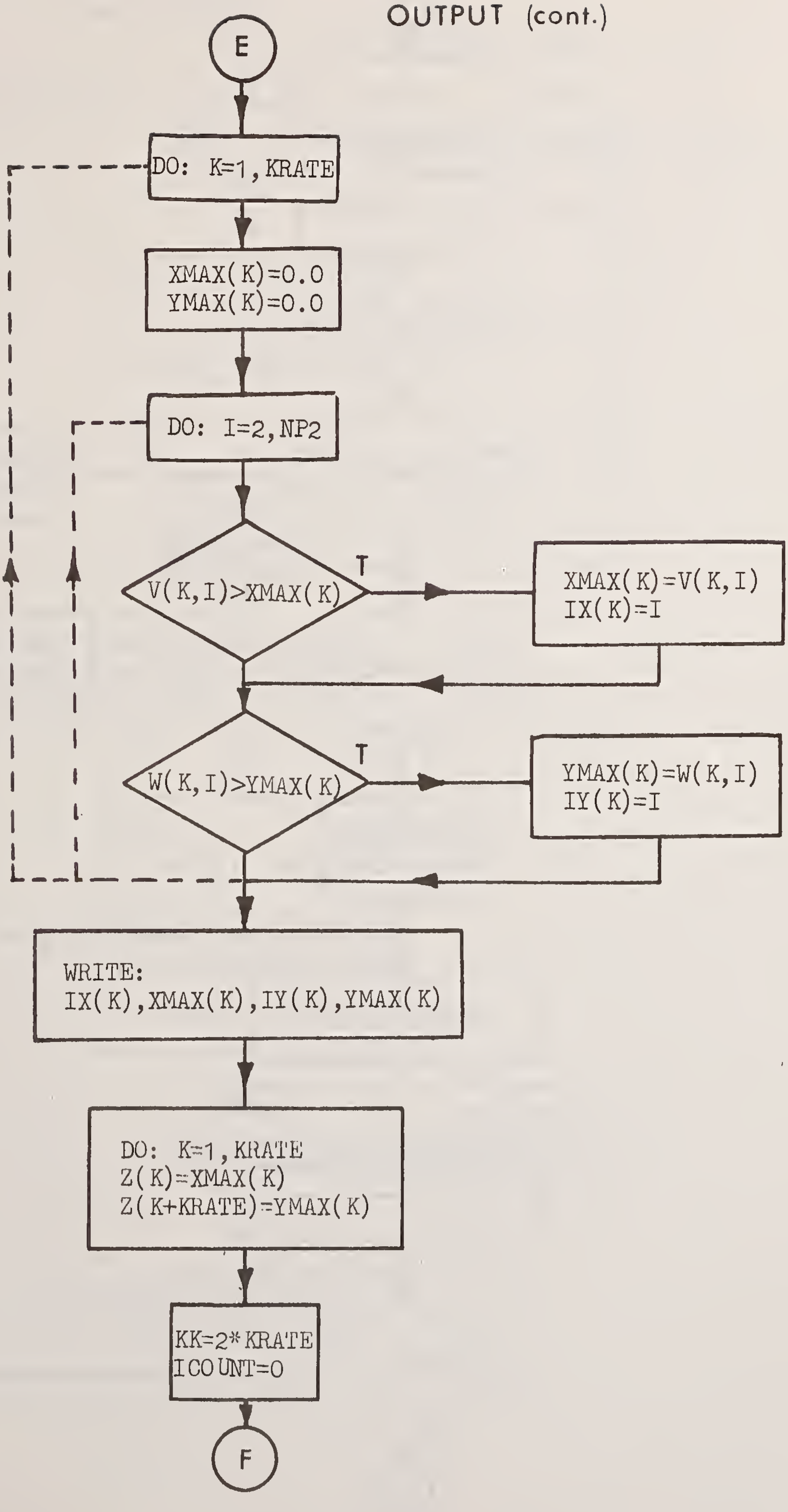
OUTPUT (cont.)

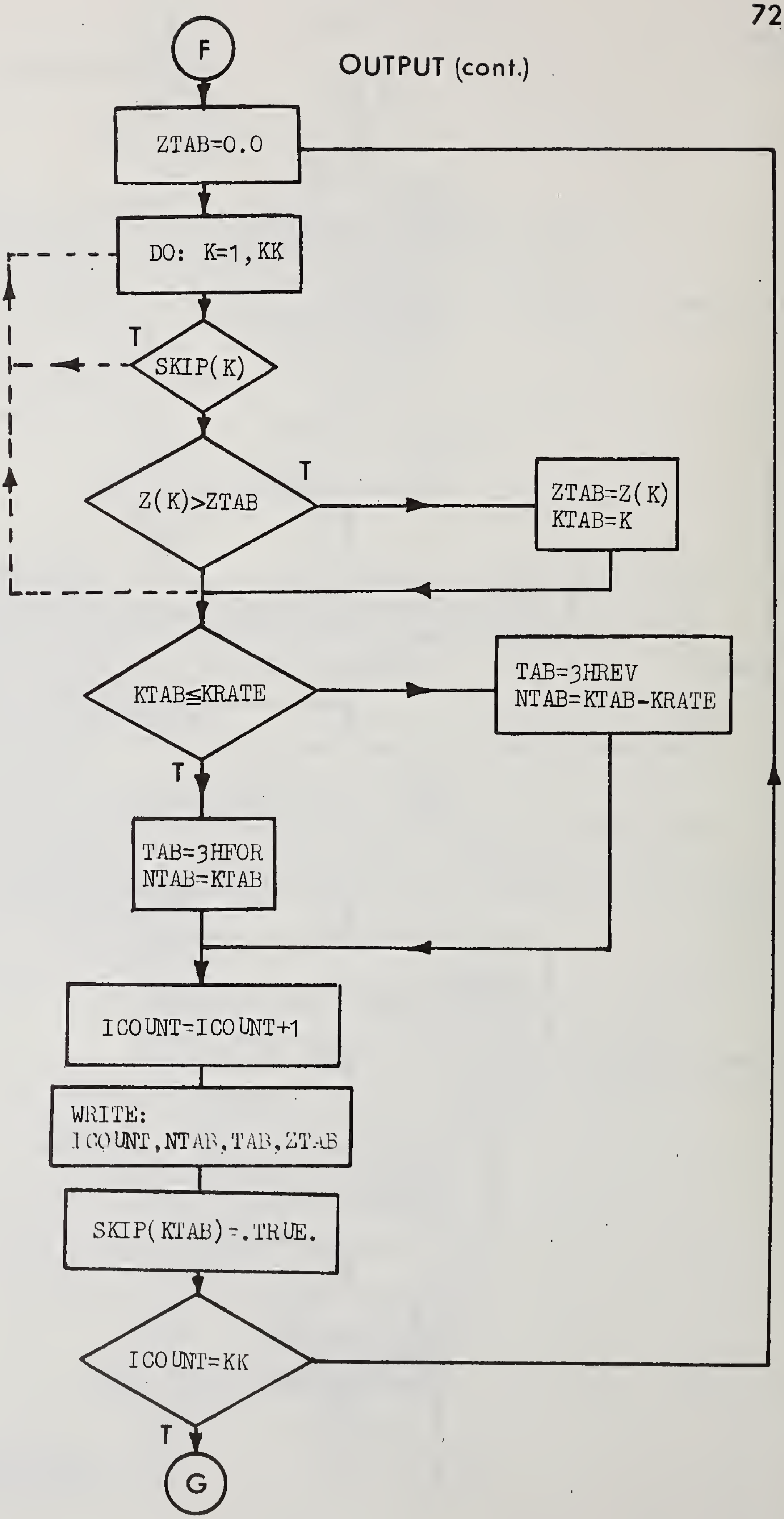

C
ICOUNT=0
ISTART=4
ISTOP=ISTART+15
DO: K=1,KRATE
DO: I=ISTART,ISTOP
K=5
T
FOR(I)=RATE(K,I)*F(LA(K),I)*F(LB(K),I)*FS(JBODYA,I)*RH2(I)
REV(I)=RATER(K,I)*F(LRA(K),I)*FS(JBODYA,I)*RHO(I)
SFAC=PUT(LRA(K))-PUT(LA(K))-PUT(LB(K))
HV(K,I)=SFAC*FOR(I)
HW(K,I)=-SFAC*REV(I)
K=10
T
FOR(I)=RATE(K,I)*F(LA(K),I)*F(LB(K),I)*FS(JBODYB,I)*RH2(I)
REV(I)=RATER(K,I)*F(LRA(K),I)*FS(JBODYB,I)*RHO(I)
SFAC=PUT(LRA(K))-PUT(LA(K))-PUT(LB(K))
HV(K,I)=SFAC*FOR(I)
HW(K,I)=-SFAC*REV(I)
d
dd
add
D
dddd

OUTPUT (cont.)

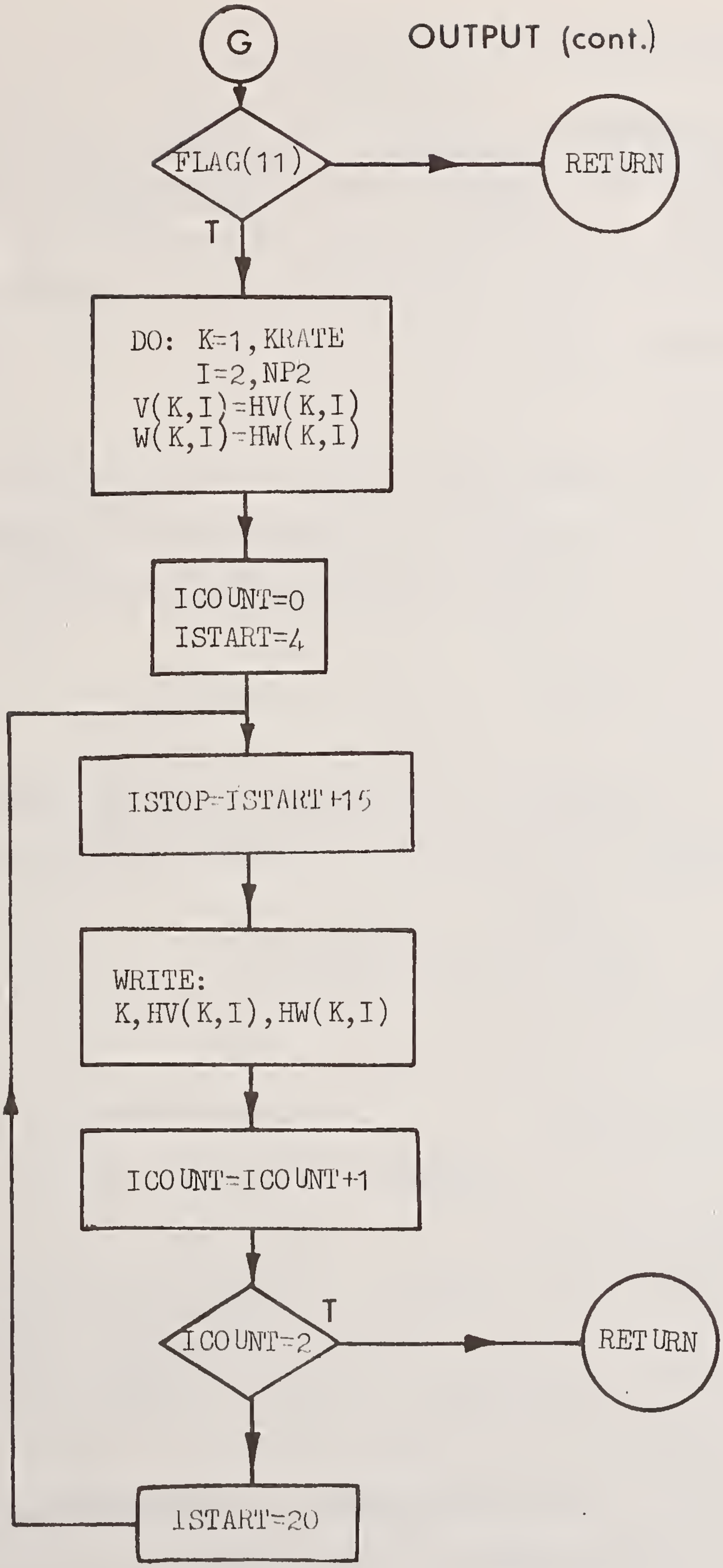

d
dd
ddd
D
dddd
BOD1(K)
T
FOR(I)=RATE(K,I)*F(LA(K),I)*F(LB(K),I)*RHO(I)
REV(I)=RATER(K,I)*F(LRA(K),I)*F(LRB(K),I)*RHO(I)
SFAC=PUT(LRA(K))+PUT(LRB(K))-PUT(LA(K))-PUT(LB(K))
HV(K,I)=SFAC*FOR(I)
HW(K,I)=-SFAC*REV(I)
BOD2(K)
T
FOR(I)=RATE(K,I)*F(LA(K),I)*F(LB(K),I)*FS(JMR,I)*RH2(I)
REV(I)=RATER(K,I)*F(LRA(K),I)*FS(JMR,I)*RHO(I)
SFAC=PUT(LRA(K))-PUT(LA(K))=PUT(LB(K))
HV(K,I)=SFAC*FOR(I)
HW(K,I)=-SFAC*REV(I)
FOR(I)=RATE(K,I)*F(LA(K),I)*FS(JMR,I)*RHO(I)
REV(I)=RATER(K,I)*F(LRA(K),I)*F(LRB(K),I)*FS(JMR,I)*RH2(I)
SFAC=PUT(LRA(K))+PUT(LRB(K))=PUT(LA(K))
HV(K,I)=SFAC*FOR(I)
HW(K,I)=-SFAC*REV(I)
V(K,I)=FOR(I)
W(K,I)=REV(I)
WRITE:
K,FOR(I),REV(I)
ICOUNT=ICOUNT+1
45
ICOUNT=2
T
ISTART=20
E

OUTPUT (cont.)


E
DO: K=1,KRATE
XMAX(K)=0.0
YMAX(K)=0.0
DO: I=2,NP2
V(K,I)>XMAX(K)
T
XMAX(K)=V(K,I)
IX(K)=I
W(K,I)>YMAX(K)
T
YMAX(K)=W(K,I)
IY(K)=I
WRITE:
IX(K),XMAX(K),IY(K),YMAX(K)
DO: K=1,KRATE
Z(K)=XMAX(K)
Z(K+KRATE)=YMAX(K)
KK=2*KRATE
ICOUNT=0
F


F
OUTPUT (cont.)
ZTAB=0.0
DO: K=1,KK
T
SKIP(K)
T
Z(K)>ZTAB
ZTAB=Z(K)
KTAB=K
KTAB≤KRATE
TAB=3HREV
NTAB=KTAB-KRATE
T
TAB=3HFOR
NTAB=KTAB
ICOUNT=ICOUNT+1
WRITE:
ICOUNT,NTAB,TAB,ZTAB
SKIP(KTAB)=.TRUE.
ICOUNT=KK
T
G

OUTPUT (cont.)


G
FLAG(11)
RETURN
T
DO: K=1,KRATE
I=2,NP2
V(K,I)=HV(K,I)
W(K,I)=HW(K,I)
ICOUNT=0
ISTART=4
ISTOP=ISTART+15
WRITE:
K,HV(K,I),HW(K,I)
ICOUNT=ICOUNT+1
ICOUNT=2
T
RETURN
ISTART=20

IV. TEST CASE

As an example of the use of this program we have utilized the H_2-Br_2 flame. The input data were taken from a paper by Lovachev and Kaganova.[2] Spalding and Stephenson[3] have also used the Spalding method for this flame. The following input information was used:

Thermal conductivity:

$$\lambda = \lambda_u (T/T_u)^{0.67}$$

where λ_u is the thermal conductivity at the temperature T_u of the unburnt gas. It is assumed to be independent of concentration. For $T_u = 323^\circ K$, λ_u is taken to have the value 3.349×10^{-2} J/m-$^\circ$K-s.

Diffusion Coefficients:

$$\Delta_j = \Delta_{ju} (T/T_u)^{1.67}/P$$

where P is the pressure in atmospheres and the Δ_{ju} are considered to be independent of concentration and are the values at temperature T_u. For $T_u = 323^\circ K$, we use

$$\Delta_{H_2u} = \Delta_{Br_2u} = \Delta_{HBru} = 0.101 \times 10^{-4} m^2/s$$

$$\Delta_{Hu} = 1.05 \times 10^{-4} m^2/s$$

$$\Delta_{Bru} = 0.155 \times 10^{-4} m^2/s$$

Thermodynamic properties:

The specific heat of the mixture is assumed to be independent of temperature and concentration. The value used was $C_p = 532 J/kg$-$^\circ K = d_j$. In this case, the coefficients e_j and $f_j = 0$ and all species have the same value of d_j.

The enthalpies of formation as taken from the JANAF tables are,

$h^R_{Br}(298^\circ) = 1.4009 \times 10^6 J/kg$ $\qquad$ $h^R_{Br_2}(298^\circ) = 1.93505 \times 10^5 J/kg$

$h^R_{H}(298^\circ) = 2.1641 \times 10^8$ $\qquad$ $h^R_{HBr}(298^\circ) = -4.4807 \times 10^5$

$h^R_{H_2}(298^\circ) = 0.0$

Note that the value for HBr is slightly different from the value of -4.5304×10^5 J/kg used by Lovachev and Kaganova.
For the program we want values for h_j^o = HREF(J) = $h_j^R - d_j T_R$, where T_R is the reference temperature for h_j^R. We have

$$h^o_{Br} = 1.242540\times10^6 \text{ J/kg}$$

$$h^o_H = 2.162510\times10^8$$

$$h^o_{H_2} = -1.583691\times10^5$$

$$h^o_{Br_2} = 3.513587\times10^4$$

$$h^o_{HBr} = -6.064374\times10^5$$

Rate Constants:

The reaction scheme used is

		ΔH^o
$Br_2 + M \rightleftarrows Br + Br + M$	(1)	46.09 kcal/mole
$H_2 + M \rightleftarrows H + H + M$	(2)	104.20
$Br + H_2 \rightleftarrows HBr + H$	(3)	16.70
$H + Br_2 \rightleftarrows HBr + Br$	(4)	-41.41

The forward rate constants used were

$$k_1 = 7.03\times10^{11} T^{-\frac{1}{2}} \exp(-2.3655\times10^4/T) \quad m^3/\text{mole-s}$$

$$k_2 = 7.63\times10^{13} T^{-1} \exp(-5.4356\times10^4/T)$$

$$k_3 = 3.46\times10^4 T \exp(-8.3547\times10^3/T)$$

$$k_4 = 6.42\times10^6 T^{\frac{1}{2}} \exp(-5.5866\times10^2/T)$$

Free energies of reaction in kcal/mole were calculated from the formulas,

$$\Delta F^o_1 = 46.1756 - 2.49467\times10^{-2} T - 6.5671\times10^{-7} T^2$$

$$\Delta F^o_2 = 104.426 - 2.38954\times10^{-2} T - 1.417614\times10^{-6} T^2$$

$$\Delta F^o_3 = 16.532 - 1.49615\times10^{-3} T - 1.91112\times10^{-7} T^2$$

$$\Delta F^o_4 = -41.7184 - 2.54835\times10^{-3} T + 5.69792\times10^{-7} T^2$$

For the reverse reactions Lovachev and Kaganova use

$k_1' = 3.63 \times 10^3 \ m^6/mole^2\text{-}s$

$k_2' = 3.63 \times 10^3$

$k_3' = 9.06 \times 10^5 T^{\frac{1}{2}} \exp(-8.6064 \times 10^2/Y) \ m^3/mole\text{-}s$

$k_4' = 6.52 \times 10^4 T \exp(-2.1440 \times 10^4/T)$

These formulas yield values of the reverse rate constants which are slightly different from those calculated with the ΔF^o values via the equilibrium constants. For comparison, Table I gives the forward rates, equilibrium constants K_p, reverse rates calculated from K_p, and the reverse rates of Lovachev and Kaganova.

The other conditions for this test case were, pressure = 1 atm, cold boundary temperature = 323^oK, cold gas composition, Br_2 = 40mole%, balance, H_2.

The format for this input data is shown in Table II. Cards containing the symbol @ are control statements for the NBS UNIVAC 1108.

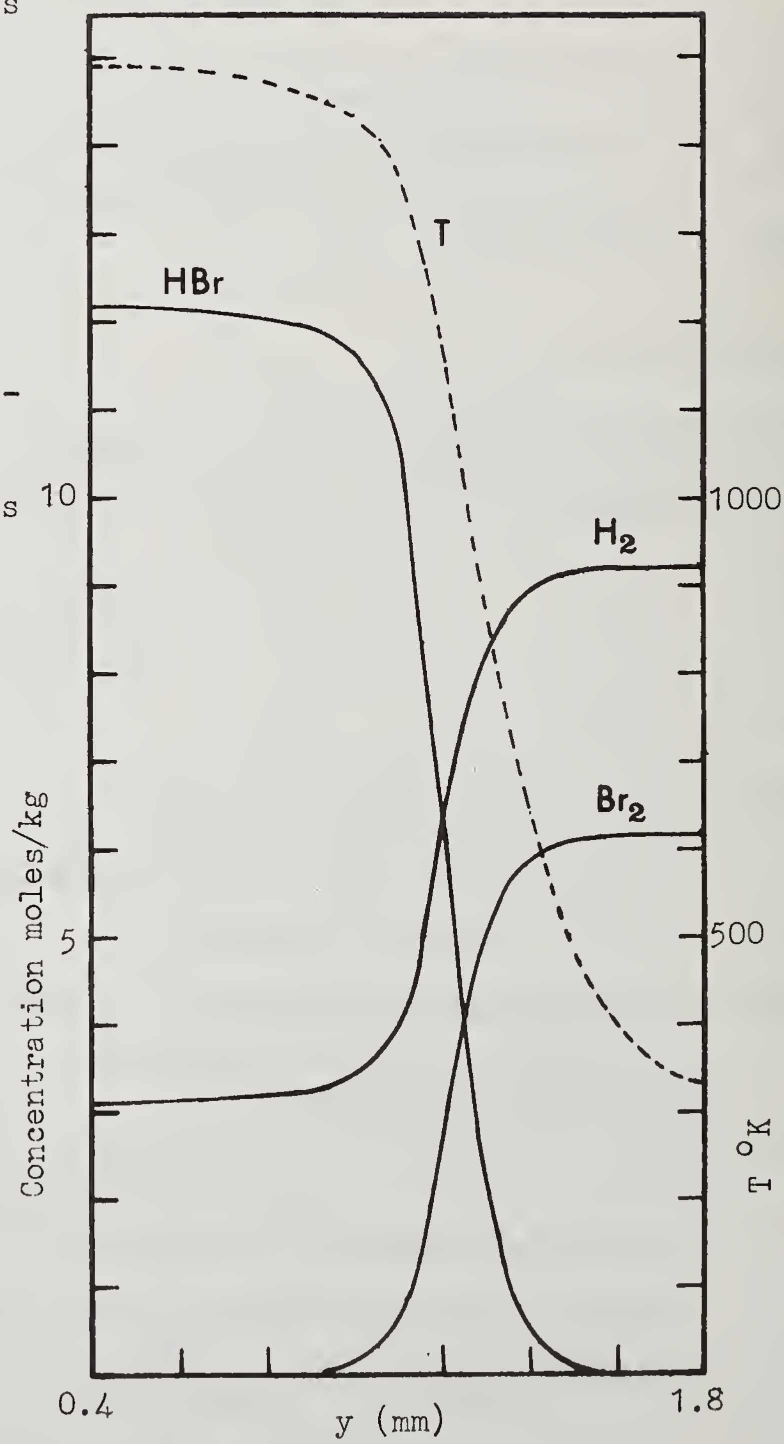


Figure 4
Concentration Profiles for the Major Species

TABLE I. Rate constants for the H_2-Br_2 flame as a function of temperature.

Temperature	k_1	K_p	k'_1	k'^{a}_1
400	7.2918×10^{-16}	1.8963×10^{-20}	1.2621×10^{3}	3.6300×10^{3}
600	2.1668×10^{-7}	5.2174×10^{-12}	2.0447×10^{3}	"
800	3.5799×10^{-3}	8.9451×10^{-8}	2.6272×10^{3}	"
1000	1.1850	3.1882×10^{-5}	3.0499×10^{3}	"
1200	5.5762×10^{1}	1.6384×10^{-3}	3.3513×10^{3}	"
1400	8.6271×10^{2}	2.7841×10^{-2}	3.5598×10^{3}	"
1600	6.6700×10^{3}	2.3689×10^{-1}	3.6967×10^{3}	"

Temperature	k_2	K_p	k'_2	k'^{a}_2
400	1.8373×10^{-48}	1.9216×10^{-52}	3.1383×10^{2}	3.6300×10^{3}
600	5.7569×10^{-29}	2.3267×10^{-33}	1.2182×10^{3}	"
800	2.9600×10^{-19}	8.6949×10^{-24}	2.2348×10^{3}	"
1000	1.8881×10^{-13}	5.1000×10^{-18}	3.0379×10^{3}	"
1200	1.3529×10^{-9}	3.7477×10^{-14}	3.5547×10^{3}	"
1400	7.4921×10^{-7}	2.2544×10^{-11}	3.8178×10^{3}	"
1600	8.4011×10^{-5}	2.8378×10^{-9}	3.8868×10^{3}	"

Temperature	k_3	K_p	k'_3	k'^{a}_3
400	1.1753×10^{-2}	2.0411×10^{-9}	5.7582×10^{6}	2.1073×10^{6}
600	1.8616×10^{1}	2.1356×10^{-6}	8.7170×10^{6}	5.2875×10^{6}
800	8.0661×10^{2}	6.9745×10^{-5}	1.1565×10^{7}	8.7390×10^{6}
1000	8.1410×10^{3}	5.6918×10^{-4}	1.4303×10^{7}	1.2116×10^{7}
1200	3.9318×10^{4}	2.3220×10^{-3}	1.6933×10^{7}	1.5319×10^{7}
1400	1.2402×10^{5}	6.3739×10^{-3}	1.9457×10^{7}	1.8332×10^{7}
1600	2.9885×10^{5}	1.3658×10^{-2}	2.1881×10^{7}	2.1163×10^{7}

Temperature	k_4	K_p	k'_4	k'^{a}_4
400	3.1769×10^{7}	2.0142×10^{23}	1.5773×10^{-16}	1.3729×10^{-16}
600	6.1978×10^{7}	4.7888×10^{15}	1.2942×10^{-8}	1.1838×10^{-8}
800	9.0324×10^{7}	7.1752×10^{11}	1.2588×10^{-4}	1.1968×10^{-4}

TABLE V. Continued.

Temperature	k_4	K_p	k'_4	k'_4[a]
1000	1.1612×10^{8}	3.5582×10^{9}	3.2634×10^{-2}	3.1826×10^{-2}
1200	1.3962×10^{8}	1.0152×10^{8}	1.3752	1.3611
1400	1.6117×10^{8}	7.8715×10^{6}	2.0475×10^{1}	2.0386×10^{1}
1600	1.8112×10^{8}	1.1402×10^{6}	1.5885×10^{2}	1.5802×10^{2}

[a]Reverse rate constants calculated from the expressions given by Lovachev and Kaganova.

TABLE II. Format of input data for the H_2-Br_2 flame.

```
@RUN,M/R BROWN2,36242-BROWNR,ABCDE,15,75
@ASG,A FLAME.
@FRT,T FLAME.
@MAP,IS TEST,TEST
LIB FLAME.
IN SPALD/3,INITL/1,TRANS/6,RATCN/3,SCHM2/7,CALC/1
IN ENTRN/3,SENTP/1,SCHM1/1,ZCALC/1
END
@XQT TEST

5 DEC 74 R1 H2-BR2 FLAME. LOVACHEV & KAGNOVA DATA.          | RUNID(M),M=1,12
   BR       H        H2       BR2      HBR    BLANK    BLANK | HEADNG(M),M=1,126
  BLANK    BLANK    BLANK    BLANK    BLANK  ENTHALPY  TEMP  |
 1 2 3 4 5 6 7 8 9101112                                    | LH,LO,LOH,LHO2,LH202,LX,LHX,LX2,LO2,
                                                            |                    LH20,LH2,LN2
301530                                                      | LD,LE,LC
051001                                                      | LDELA,LDELB,LDELC
3008231.3D0                                                 | N,NII,NEE,OMR
 6 2 1 2 3 4 2 0 4 5 6                                      | JH,JRAD,JM,JMR,JT,JTR,ITER,ITER1,
 .200D0 .000D0 .800D0                                       | XI(J),J=JRADP1,JHM1          JENTRN,
 .600D0 .400D0 .000D0                                       | XE(J),J=JRADP1,JHM1          JBODYA,
001                                                         | INPRFL(J),J=JRADP1,JHM1      JBODYB
323.D01.00D0.2D-2                                           | TCOLD,PRESS,YWIDTH
 1.242540D+6 0.531724D+3 0.000000D+0 0.000000D+0 79.916D-3  | HREF(J),DCP(J),ECP(J),FCP(J),WT(J),
 2.162510D+8 0.531724D+3 0.000000D+0 0.000000D+0  1.008D-3  |                   J=1,JHM1
-1.583691D+5 0.531724D+3 0.000000D+0 0.000000D+0  2.016D-3  |
 3.513587D+4 0.531724D+3 0.000000D+0 0.000000D+0 159.83D-3  |
-6.064374D+5 0.531724D+3 0.000000D+0 0.000000D+0  80.92D-3  |
5.D-65.D-4600.D0                                            | DXMIN,DXMAX,STEPS
300030                                                      | IMAX,LPRINT
.10                                                         | ALPHA
04                                                          | KRAT
7.0D11-.500D02.3655D4TT                                     | FRQ(K),BETA(K),EACT(K),BFLAG(K),
7.6D13-1.00D05.4356D4TT                                     |           EFLAG(K),K=1,KRATE
3.46D4 1.00D08.3547D3TT                                     |
6.42D6 .500D05.5866D2TT                                     |
```

TABLE II. Continued.

```
46.17560D+0-2.49476D-2-6.56710D-7TT           EQA(K),EQB(K),EQC(K),BOD1(K),BOD2(K),
104.4260D+0-2.38954D-2-1.41761D-6TT                                 K=1,KRATE
16.53200D+0-1.49615D-3-1.91112D-7FF
-41.7184D+0-2.54835D-3 5.69792D-7FF
0000000000                                    ITEST(M),M=1,10
00                                            INPUT
 1 2 3 4                                      LORDER(J),J=1,JHM2
TTTT                                          SPECIE(J),J=1,JHM2
TTTT                                          REACT(K),K=1,KRATE
TTTT                                          FFLAG(K),K=1,KRATE
TTTT                                          RFLAG(K),K=1,KRATE
 4 0 1 1                                      LA(K),LB(K),LRA(K),LRB(K),K=1,KRATE
 3 0 2 2
 1 3 5 2
 2 4 5 1
FTFTTFFTTTTFFFFFFFFF                          FLAG(I),I=1,20

@EOF
@FIN
```

Since the format of the output makes it unsuitable for reproduction in this report, we shall illustrate some of it in graphic form. The steady-state velocity of this flame is 0.241 m/s and the hot side flame temperature is 1518°K for a complete reaction. This velocity is 1.2% lower than that obtained by Spalding and Stephenson; the difference probably arises from the slightly different values used here for the reverse rate constants. (See Table I.)

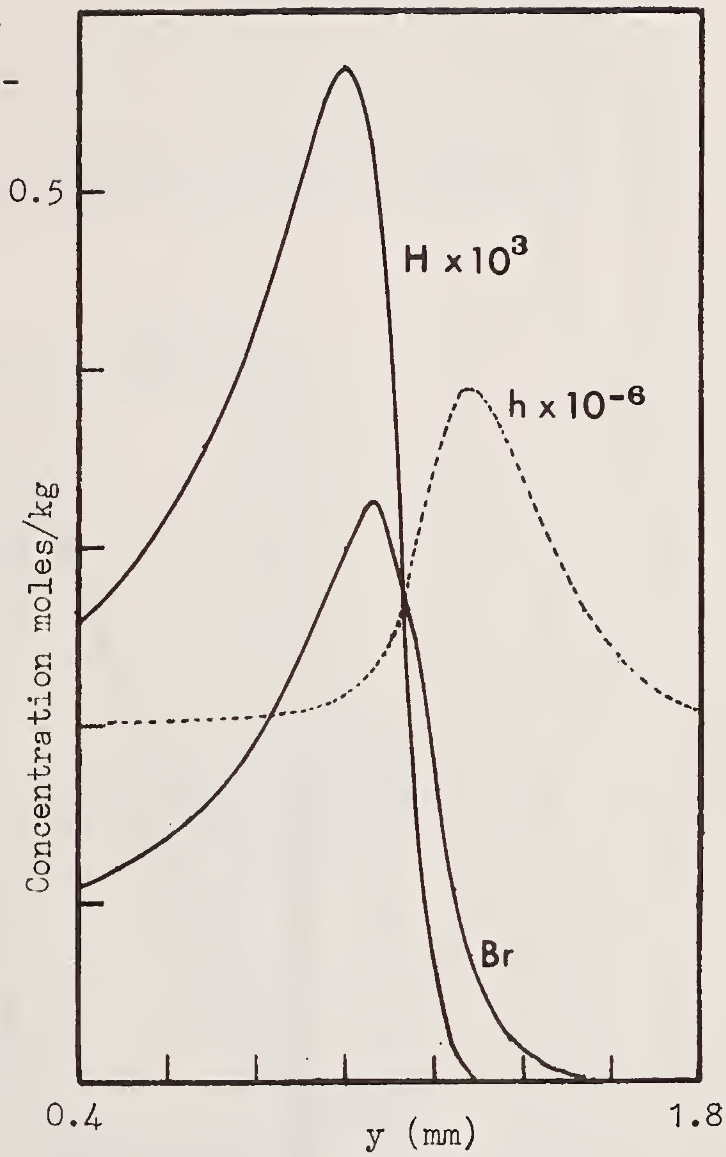


Figure 5
Concentration Profiles for the Minor Species

Concentration profiles for the major species are shown in Fig. 4. These are plotted as a function of the laboratory spatial variable y in mm from the hot boundary. Also shown here is the temperature profile.

Profiles for the two trace species, H and Br, are shown in Fig. 5 along with the enthalpy profile. This latter profile has units of J/kg. The rates of the major reactions are shown in Fig. 6. The + sign in the figure denotes a forward rate. Figure 7 gives the heat release rates for the major reactions and also the total heat release rate as a function of distance through the flame.

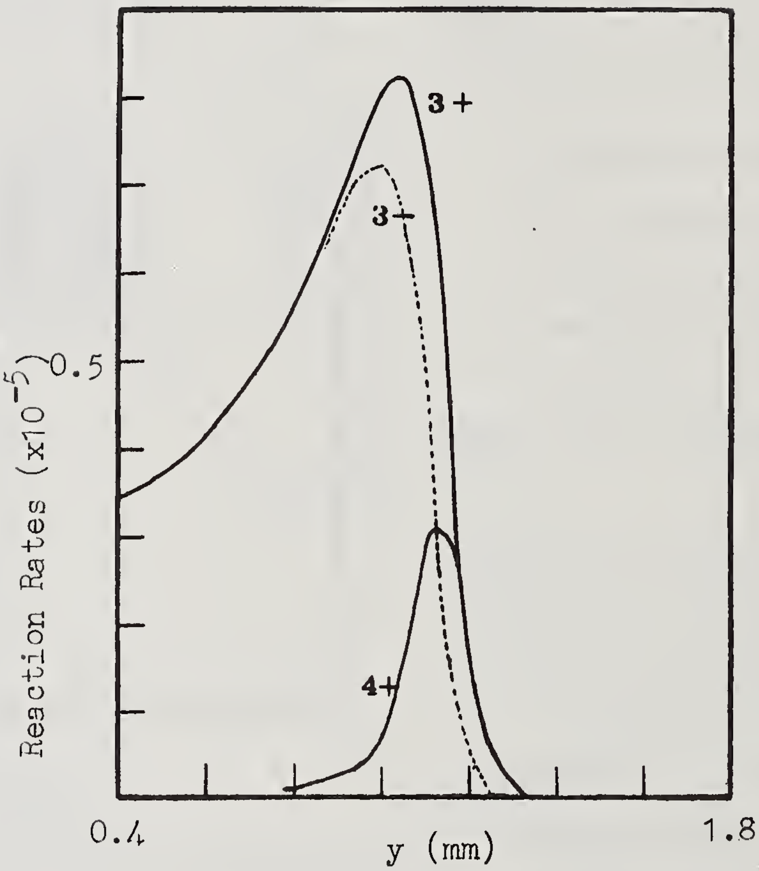


Figure 6
Rates of Major Reactions

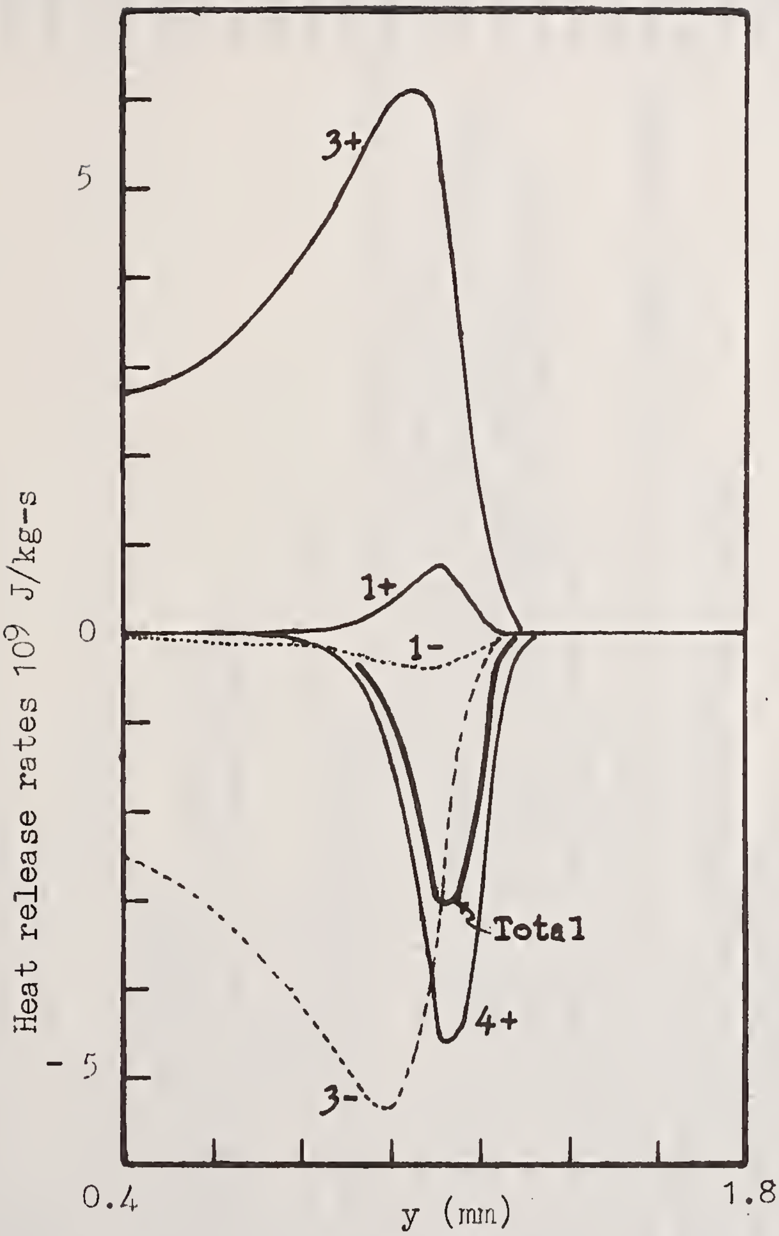


Figure 7
Heat Release Rates

```
@ELT,SIB ABCDE*FLAME(2).SPALD/3,,,173314133212
      IMPLICIT REAL*8 (A-H,Ø-Z)
      REAL ALPHA,RUNID(12),HEADNG(126)
      LØGICAL BØD1,BØD2,FLAG,G1,G2,G3
      LØGICAL SPECIE(15),REACT,BFLAG,EFLAG,FFLAG,RFLAG
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/ØMEGA1/NII,NEE,ØMR
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)
      CØMMØN/ØMEGA3/BØM(43),BØMT3(43),ØMP(43),PEIØMD(43),PEIØM2(43)
      CØMMØN/INIT/XI(15),XE(15),IPRFL(15),TCØLD,PRESS,YWIDTH
      CØMMØN/TRANS1/HREF(15),DCP(15),ECP(15),FCP(15),WT(15),ITER,ITER1
      CØMMØN/TRANS2/RMB(43),PREF(15,43),HDRM(15,43),TRCP(15,43)
     *,DR2M(15,43)
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)
      CØMMØN/TRANS4/DA(  14,14),DB(14,14),TDIV(14,14)
      CØMMØN/VAR1/F(15,43),FS(15,43)
      CØMMØN/YCALC/PEI,Y(43),YD(43),YAV(43),RPEI
      CØMMØN/RTPARM/FRQ(30),BETA(30),EACT(30)
     * ,BØD1(30),BØD2(30),REACT(30)
      CØMMØN/RATC/RATE(30,43),RATER(30,43),KRATE,EQA(30),EQB(30),EQC(30)
      CØMMØN/SCM1/SC1(15,43)
      CØMMØN/SCM2/SU(15,43),SD(15,43)
      CØMMØN/SENT/SEN(43)
      CØMMØN/CØEF/A(15,43),B(15,43),C(15,43),D(15,43)
      CØMMØN/ENTR/VEL(15),RMI,RME,DX,ALPHA
      CØMMØN/MASPRØ/R(15,43)
      CØMMØN/CMINDX/LH,LØ,LØH,LHØ2,LH2Ø2,LX,LHX,LX2,LØ2,LH2Ø,LH2,LN2
      CØMMØN/LVALS/LCØUNT,LPRINT
      CØMMØN/REACT/LA(30),LB(30),LRA(30),LRB(30)
      CØMMØN/FLAGS/FLAG(20)
      CØMMØN/FLAGR/BFLAG(30),EFLAG(30),FFLAG(30),RFLAG(30)
      CØMMØN/GSWITC /G1(30,15),G2(30,15),G3(30,15)
      DIMENSIØN ITEST(10),PGØMP(43),PGØMM(43),PBØM(43)
      DIMENSIØN LØRDER(15)
```

```
      DATA SC1/645*.0D0/,SEN/43*.0D0/,PREF/645*.7D-5/,RMB/43*.0D0/,      SPD00350
     * HDRM/645*.0D0/,TRCP/645*.0D0/,DR2M/645*.0D0/                      SPD00360
C                                                                        SPD00370
C     A.  PARAMETERS AND CØNTRØL INDICES.                                SPD00380
      READ(5,2000) (RUNID(M),M=1,12)                                     SPD00390
 2000 FØRMAT(12A6)                                                       SPD00400
      READ(5,2001)(HEADNG(M),M=1,126)                                    SPD00410
 2001 FØRMAT(63A1)                                                       SPD00420
      READ(5,2002)LH,LØ,LØH,LHØ2,LH2Ø2,LX,LHX,LX2,LØ2,LH2Ø,LH2,LN2       SPD00430
 2002 FØRMAT(14I2)                                                       SPD00440
      READ(5,450) LD,LE,LC                                               SPD00450
      READ(5,450) LDELA,LDELB,LDELC                                      SPD00460
  450 FØRMAT(3I2)                                                        SPD00470
      LTABA=0                                                            SPD00480
      LTABB=0                                                            SPD00490
      LTABC=0                                                            SPD00500
      READ(5,500) N,NII,NEE,ØMR                                          SPD00510
  500 FØRMAT(3I2,D5.1)                                                   SPD00520
      NP1=N+1                                                            SPD00530
      NP2=N+2                                                            SPD00540
      NP3=N+3                                                            SPD00550
      READ(5,503)JH,JRAD,JM,JMR,JT,JTR,ITER,ITER1,JENTRN,JBØDYA,JBØDYB   SPD00560
  503 FØRMAT(11I2)                                                       SPD00570
      JHM1=JH-1                                                          SPD00580
      JHM2=JH-2                                                          SPD00590
      JHM3=JH-3                                                          SPD00600
      JRADP1=JRAD+1                                                      SPD00610
      READ(5,502) (XI(J),J=JRADP1,JHM1)                                  SPD00620
      READ(5,502) (XE(J),J=JRADP1,JHM1)                                  SPD00630
  502 FØRMAT(5D7.3)                                                      SPD00640
      READ(5,103) (IPRFL(J),J=JRADP1,JHM1)                               SPD00650
  103 FØRMAT(5I1)                                                        SPD00660
      READ(5,101) TCØLD,PRESS,YWIDTH                                     SPD00670
  101 FØRMAT(D6.0,D6.2,D5.1)                                             SPD00680
      READ(5,504) (HREF(J),DCP(J),ECP(J),FCP(J),WT(J),J=1,JHM1)          SPD00690
```

```
  504 FØRMAT( 4D12.6, D10.2 )
      READ( 5, 506 ) DXMIN, DXMAX, STEPS
  506 FØRMAT( 2D5.0, D6.0 )
      FAC2=( DXMAX-DXMIN )/STEPS
      READ( 5, 508 ) LMAX, LPRINT
  508 FØRMAT( 2I3 )
      READ( 5, 510 ) ALPHA
  510 FØRMAT( F3.2 )
      READ( 5, 512 ) KRAT
  512 FØRMAT( I2 )
      KRATE=KRAT
      READ( 5, 514 ) ( FRQ( K ), BETA( K ), EACT( K ), BFLAG( K ), EFLAG( K ), K=1, KRATE )
  514 FØRMAT( D6.2, D7.2, D8.2, 2L1 )
      READ( 5, 515 ) ( EQA( K ), EQB( K ), EQC( K ), BØD1( K ), BØD2( K ), K=1, KRATE )
  515 FØRMAT( 3D11.5, 2L1 )
      DØ 517 K=1, KRATE
      EQA( K )=EQA( K )*5.03297D2
      EQB( K )=EQB( K )*5.03297D2
  517 EQC( K )=EQC( K )*5.03297D2
      READ( 5, 516 ) ( ITEST( M ), M=1, 10 )
  516 FØRMAT( 10I1 )
      READ( 5, 512 ) INPUT
      READ( 5, 2002 ) ( LØRDER( J ), J=1, JHM2 )
      READ( 5, 2003 ) ( SPECIE( J ), J=1, JHM2 )
 2003 FØRMAT( 14L1 )
      READ( 5, 2005 ) ( REACT( K ), K=1, KRATE )
 2005 FØRMAT( 30L1 )
      READ( 5, 2005 ) ( FFLAG( K ), K=1, KRATE )
      READ( 5, 2005 ) ( RFLAG( K ), K=1, KRATE )
      READ( 5, 354 ) ( LA( K ), LB( K ), LRA( K ), LRB( K ), K=1, KRATE )
  354 FØRMAT( 4I2 )
      READ( 5, 2005 ) ( FLAG( I ), I=1, 20 )
      CALL SWITCH( KRATE )
      IF( .NØT.FLAG( 1 )) CALL DIFUSE( PRESS )
      JSTØP=JHM3
```

```
      IF(FLAG(2)) JSTØP=JHM2                                            SPD01050
      IF(INPUT.EQ.1) READ(45) ((F(J,I),J=1,JH),I=1,NP3),PEI             SPD01060
      L=0                                                               SPD01070
      LCØUNT=0                                                          SPD01080
      X=.0D0                                                            SPD01090
      DX=DXMIN                                                          SPD01100
C                                                                       SPD01110
C                                                                       SPD01120
C     B.  GRID PRØPERTIES,CRØSS STREAM DISTANCE,RATE CØNSTANTS,AND      SPD01130
C       TRANSPØRT PRØPERTIES.                                           SPD01140
C                                                                       SPD01150
C     CALCULATIØN ØF GRID.                                              SPD01160
      CALL ØMEGA                                                        SPD01170
C                                                                       SPD01180
C     ---------------------- TEST 1 ------------------------------------SPD01190
      IF(ITEST(1).EQ.1) GØ TØ 700                                       SPD01200
      GØ TØ 702                                                         SPD01210
  700 WRITE(6,650)                                                      SPD01220
  650 FØRMAT(1X,7HNII,NEE)                                              SPD01230
      WRITE(6,600) NII,NEE                                              SPD01240
  600 FØRMAT(1X,2I3)                                                    SPD01250
      WRITE(6,604)                                                      SPD01260
  604 FØRMAT(1H )                                                       SPD01270
      WRITE(6,651)                                                      SPD01280
  651 FØRMAT(1X,35HØM(I),ØMD(I),BØM(I),BØMT3(I),ØMP(I))                 SPD01290
      WRITE(6,601)(ØM(I),ØMD(I),BØM(I),BØMT3(I),ØMP(I),I=1,NP3)         SPD01300
  601 FØRMAT(1X,5D14.5)                                                 SPD01310
      WRITE(6,604)                                                      SPD01320
      WRITE(6,652)                                                      SPD01330
  652 FØRMAT(1X,7HØMI,ØME)                                              SPD01340
      WRITE(6,602) ØMI,ØME                                              SPD01350
  602 FØRMAT(1X,2D14.5)                                                 SPD01360
C     ------------------------------------------------------------------SPD01370
C                                                                       SPD01380
C     CALCULATIØN ØF INITIAL CØNDITIØNS/ ØR IF INPUT=1,THEY ARE READ IN.SPD01390
```

```
  702 IF( INPUT.EQ.0) CALL INITL                                         SPD01400
C     ---------------------- TEST 2 ------------------------------------SPD01410
      IF(ITEST(2).EQ.1) GØ TØ 704                                        SPD01420
      GØ TØ 706                                                          SPD01430
  704 WRITE(6,604)                                                       SPD01440
      WRITE(6,604)                                                       SPD01450
      WRITE(6,653)                                                       SPD01460
  653 FØRMAT(1X,34H((F(J,I),J=1,JH),FS(JT,I),I=1,NP3))                   SPD01470
      WRITE(6,606) ((F(J,I),J=1,JH),FS(JT,I),I=1,NP3)                    SPD01480
  606 FØRMAT(1X,7D14.5)                                                  SPD01490
C     ------------------------------------------------------------------SPD01500
C                                                                        SPD01510
C     TRANSPØRT PRØPERTIES, DENSITY, AND TEMPERATURE.                    SPD01520
  706 IF(L.LT.LD) GØ TØ 400                                              SPD01530
      LTABA=LTABA+1                                                      SPD01540
      IF(LTABA.EQ.LDELA) GØ TØ 401                                       SPD01550
      CALL TRANS(0)                                                      SPD01560
      GØ TØ 402                                                          SPD01570
  401 LTABA=0                                                            SPD01580
  400 CALL TRANS(1)                                                      SPD01590
  402 CØNTINUE                                                           SPD01600
      IF(L.GT.0) GØ TØ 31                                                SPD01610
      IF(INPUT.EQ.0) GØ TØ 30                                            SPD01620
      GØ TØ 31                                                           SPD01630
   30 PEI=1.D0                                                           SPD01640
      CALL ZCALC                                                         SPD01650
      PEI=YWIDTH/Y(NP2)                                                  SPD01660
   31 CØNTINUE                                                           SPD01670
      PEID2=PEI*.5D0                                                     SPD01680
      PEIØMD(2)=PEI*ØMD(2)                                               SPD01690
      DØ 701 I=3,NP1                                                     SPD01700
      PEIØMD(I)=PEI*ØMD(I)                                               SPD01710
  701 PEIØM2(I)=PEID2*BØM(I)                                             SPD01720
C                                                                        SPD01730
```

```
C     ----------------------- TEST 3 ----------------------------------SPD01740
      IF( ITEST( 3).EQ.1 ) GØ TØ 708                                    SPD01750
      GØ TØ 710                                                         SPD01760
  708 WRITE(6,604 )                                                     SPD01770
      WRITE(6,604 )                                                     SPD01780
      WRITE(6,654 )                                                     SPD01790
  654 FØRMAT( 1X, 28H( (PREF( J, I ),J=1,JH ),I=1,NP3))                 SPD01800
      WRITE(6,608 ) ((PREF( J, I ),J=1,JH ),I=1,NP3)                    SPD01810
  608 FØRMAT( 1X,6D14.5 )                                               SPD01820
      WRITE(6,604 )                                                     SPD01830
      WRITE(6,655 )                                                     SPD01840
  655 FØRMAT( 1X,30H( (HDRM( J, I ),J=1,JHM1 ),I=1,NP3))                SPD01850
      WRITE(6,610 ) ((HDRM( J, I ),J=1,JHM1 ),I=1,NP3)                  SPD01860
  610 FØRMAT( 1X,5D14.5 )                                               SPD01870
      WRITE(6,604 )                                                     SPD01880
      WRITE(6,656 )                                                     SPD01890
  656 FØRMAT( 1X,30H( (TRCP( J, I ),J=1,JHM1 ),I=1,NP3))                SPD01900
      WRITE(6,610 ) ((TRCP( J, I ),J=1,JHM1 ),I=1,NP3)                  SPD01910
      WRITE(6,604 )                                                     SPD01920
      WRITE(6,657 )                                                     SPD01930
  657 FØRMAT( 1X,30H( (DR2M( J, I ),J=1,JHM1 ),I=1,NP3))                SPD01940
      WRITE(6,610 ) ((DR2M( J, I ),J=1,JHM1 ),I=1,NP3)                  SPD01950
      WRITE(6,604 )                                                     SPD01960
      WRITE(6,658 )                                                     SPD01970
  658 FØRMAT( 1X, 29H(RHØ( I ),RU( I ),RMB( I ),I=1,NP3))               SPD01980
      WRITE(6,612 ) (RHØ( I ),RU( I ),RMB( I ),I=1,NP3)                 SPD01990
  612 FØRMAT( 1X,3D14.5 )                                               SPD02000
      WRITE(6,604 )                                                     SPD02010
      WRITE(6,603 )                                                     SPD02020
  603 FØRMAT( 1X, 27H( FS( JM, I ),FS( JT,I ),I=1,NP3))                 SPD02030
      WRITE(6,614 ) (FS( JM, I ),FS( JT, I ),I=1,NP3)                   SPD02040
  614 FØRMAT( 1X, 2D12.6 )                                              SPD02050
C     -----------------------------------------------------------------SPD02060
C                                                                       SPD02070
```

```
C     CALCULATIØN ØF RATE CØNSTANTS.                                    SPD02080
  710 CALL RATCN                                                        SPD02090
C                                                                       SPD02100
C     ---------------------- TEST 4 ------------------------------------SPD02110
      IF(ITEST(4).EQ.1) GØ TØ 712                                       SPD02120
      GØ TØ 714                                                         SPD02130
  712 WRITE(6,604)                                                      SPD02140
      WRITE(6,604)                                                      SPD02150
      WRITE(6,659)                                                      SPD02160
  659 FØRMAT(1X,42H((RATE(K,I),RATER(K,I),K=1,KRATE),I=2,NP2))         SPD02170
      WRITE(6,618) ((RATE(K,I),RATER(K,I),K=1,KRATE),I=2,NP2)           SPD02180
  618 FØRMAT(1X,8D14.5)                                                 SPD02190
C     ------------------------------------------------------------------SPD02200
C                                                                       SPD02210
C     CALCULATIØN ØF RMI AND RME.  PEI IS ALSØ CALCULATED FØR THE       SPD02220
C     CØMING STEP.                                                      SPD02230
  714 IF(L.EQ.0) CALL SCHM2(JENTRN)                                     SPD02240
      CALL ENTRN(JENTRN,0)                                              SPD02250
      RPEI=1.D0/PEI                                                     SPD02260
C     ---------------------- TEST 5 ------------------------------------SPD02270
      IF(ITEST(5).EQ.1) GØ TØ 716                                       SPD02280
      GØ TØ 727                                                         SPD02290
  716 WRITE(6,604)                                                      SPD02300
      WRITE(6,604)                                                      SPD02310
      WRITE(6,660)                                                      SPD02320
  660 FØRMAT(1X,26HVEL(JENTRN),RMI,RME,PEI,DX)                          SPD02330
      WRITE(6,620) VEL(JENTRN),RMI,RME,PEI,DX                           SPD02340
  620 FØRMAT(1X,5D14.5)                                                 SPD02350
C     ------------------------------------------------------------------SPD02360
C     C.  PRELIMINARIES FØR CØEFFICIENTS BEFØRE SUCCESSIVE CALCULATIØN. SPD02370
C                                                                       SPD02380
C       QUANTITIES WHICH ARE THE SAME FØR ALL I AND J VALUES.           SPD02390
  727 PX=PEI/DX                                                         SPD02400
      PD4=.25D0*PX                                                      SPD02410
```

```
      G=RMI-RME
      GD4=.25D0*G
      PG=PX*G
      PGD4=.25D0*PG
      RMID2=.5D0*RMI
C
C       QUANTITIES WHICH ARE THE SAME FØR ALL J VALUES.
C
C       ---------- MID SECTIØN ----------
      DØ 3004 I=3,NP1
      PGØMP(I)=PGD4*ØMD(I)
      PGØMM(I)=PGD4*ØMD(I-1)
 3004 PBØM(I)=PX*BØM(I)
C
C       QUANTITIES WHICH DEPEND ØN I AND J  BUT NØT ØN F(J,I).
C
C       ---------- MID SECTIØN ----------
      DØ 3005 I=3,NP1
      FLP=RMID2-GD4*ØMP(I)
      FLM=RMID2-GD4*ØMP(I-1)
      AFLP=DABS(FLP)
      AFLM=DABS(FLM)
      TFLP=FLP+FLP
      TFLM=FLM+FLM
      DØ 20 J=1,JSTØP
      TP=PREF(J,I)/PEIØMD(I)
      TM=PREF(J,I-1)/PEIØMD(I-1)
      TP=TP+AFLP+DABS(TP-AFLP)
      TM=TM+AFLM+DABS(TM-AFLM)
      A(J,I)=TP-TFLP-PGØMP(I)
      B(J,I)=TM+TFLM-PGØMM(I)
   20 D(J,I)=A(J,I)+B(J,I)+PBØM(I)
      IF(.NØT.FLAG(5)) GØ TØ 3005
      TP=PREF(JH,I)/PEIØMD(I)
      TM=PREF(JH,I-1)/PEIØMD(I-1)
```

```
      TP=TP+AFLP+DABS(TP-AFLP)                                          SPD02770
      TM=TM+AFLM+DABS(TM-AFLM)                                          SPD02780
      A(JB,I)=TP-TFLP-PGØMP(I)                                          SPD02790
      B(JB,I)=TM+TFLM-PGØMM(I)                                          SPD02800
      D(JB,I)=A(JB,I)+B(JB,I)+PBØM(I)                                   SPD02810
 3005 CØNTINUE                                                          SPD02820
C                                                                       SPD02830
C     ---------------------- TEST 7 ------------------------------------SPD02840
      IF(ITEST(7).EQ.1) GØ TØ 728                                       SPD02850
      GØ TØ 729                                                         SPD02860
  728 WRITE(6,604)                                                      SPD02870
      WRITE(6,604)                                                      SPD02880
      WRITE(6,662)                                                      SPD02890
  662 FØRMAT(1X,24HA, B, AND D CØEFFICIENTS)                            SPD02900
      WRITE(6,624) ((A(J,I),J=1,JBM2),I=3,NP1)                          SPD02910
      WRITE(6,604)                                                      SPD02920
      WRITE(6,624) ((B(J,I),J=1,JBM2),I=3,NP1)                          SPD02930
      WRITE(6,604)                                                      SPD02940
      WRITE(6,624) ((D(J,I),J=1,JBM2),I=3,NP1)                          SPD02950
  624 FØRMAT(1X,4D14.5)                                                 SPD02960
C     ------------------------------------------------------------------SPD02970
C                                                                       SPD02980
C     D.    INCLUSIØN ØF THE FIRST PART ØF THE SPECIES SØURCE TERM.     SPD02990
  729 IF(L.LT.LE) GØ TØ 403                                             SPD03000
      LTABB=LTABB+1                                                     SPD03010
      IF(LTABB.EQ.LDELB) GØ TØ 404                                      SPD03020
      GØ TØ 405                                                         SPD03030
  404 LTABB=0                                                           SPD03040
  403 CALL SCHM1                                                        SPD03050
  405 CØNTINUE                                                          SPD03060
      DØ 112 J=1,JSTØP                                                  SPD03070
      DØ 110 I=3,NP1                                                    SPD03080
  110 C(J,I)=PD4*(BØMT3(I)*F(J,I)+F(J,I+1)*ØMD(I)+F(J,I-1)*ØMD(I-1))+   SPD03090
     12.D0*SC1(J,I)                                                     SPD03100
  112 CØNTINUE                                                          SPD03110
```

```
C                                                                       SPD03120
C     ---------------------- TEST 8 ----------------------------------SPD03130
      IF(ITEST(8).EQ.1) GØ TØ 723                                       SPD03140
      GØ TØ 730                                                         SPD03150
  723 WRITE(6,604)                                                      SPD03160
      WRITE(6,604)                                                      SPD03170
      WRITE(6,663)                                                      SPD03180
  663 FØRMAT(1X,8HSC1(J,I))                                             SPD03190
      WRITE(6,624) ((SC1(J,I),J=1,JHM2),I=2,NP2)                        SPD03200
      WRITE(6,604)                                                      SPD03210
      WRITE(6,664)                                                      SPD03220
  664 FØRMAT(1X,14HC CØEFFICIENTS)                                      SPD03230
      WRITE(6,624) ((C(J,I),J=1,JHM2),I=3,NP1)                          SPD03240
C     ----------------------------------------------------------------SPD03250
C                                                                       SPD03260
C     E.   SUCCESSIVE CALCULATIØNS.                                     SPD03270
  730 KHM2=JHM2                                                         SPD03280
      KHM3=JHM3                                                         SPD03290
      DØ 300 L=1,JSTØP                                                  SPD03300
      J=LØRDER(L)                                                       SPD03310
      IF(SPECIE(J)) GØ TØ 2004                                          SPD03320
      GØ TØ 300                                                         SPD03330
 2004 CØNTINUE                                                          SPD03340
      CALL SCHM2(J)                                                     SPD03350
      DØ 310 I=3,NP1                                                    SPD03360
      C(J,I)=C(J,I)+2.D0*SU(J,I)                                        SPD03370
  310 D(J,I)=D(J,I)-2.D0*SD(J,I)                                        SPD03380
C                                                                       SPD03390
C     SØLVE FØR DØWNSTREAM F(J,I) FØR SPECIES.                          SPD03400
      CALL CALC(J)                                                      SPD03410
      F(J,2)=F(J,3)                                                     SPD03420
C                                                                       SPD03430
C     CALCULATIØN ØF F(JHM1,I)                                          SPD03440
      IF(FLAG(2)) GØ TØ 353                                             SPD03450
      DØ 351 I=2,NP2                                                    SPD03460
      SUM=.0D0                                                          SPD03470
```

```
      DØ 352 K=1,KHM3
  352 SUM=SUM+F(K,I)*WT(K)
      SUM=SUM+F(JHM1,I)*WT(JHM1)
      F(JHM2,I)=(1.DO-SUM)/WT(JHM2)
  351 CØNTINUE
      GØ TØ 300
  353 DØ 340 I=2,NP2
      SUM=.0DO
      DØ 350 K=1,KHM2
  350 SUM=SUM+F(K,I)*WT(K)
      F(JHM1,I)=(1.DO-SUM)/WT(JHM1)
  340 CØNTINUE
  300 CØNTINUE
C
C     ----------------------- TEST 9 -----------------------------------
      IF(ITEST(9).EQ.1) GØ TØ 731
      GØ TØ 732
  731 WRITE(6,604)
      WRITE(6,604)
      WRITE(6,665)
  665 FØRMAT(1X,15HSU(J,I),SD(J,I))
      WRITE(6,618) ((SU(J,I),J=1,JHM2),(SD(J,I),J=1,JHM2),I=2,NP2)
      WRITE(6,604)
      WRITE(6,666)
  666 FØRMAT(1X,27HA, B, C, AND D CØEFFICIENTS)
      WRITE(6,624) ((A(J,I),J=1, JHM2),I=3,NP1)
      WRITE(6,604)
      WRITE(6,624) ((B(J,I),J=1,JHM2),I=3,NP1)
      WRITE(6,604)
      WRITE(6,624) ((C(J,I),J=1,JHM2),I=3,NP1)
      WRITE(6,604)
      WRITE(6,624) ((D(J,I),J=1,JHM2),I=3,NP1)
C     ------------------------------------------------------------------
C
```

```
C     SOLVE FOR DOWNSTREAM ENTHALPIES.                                  SPD03820
C         COEFFICIENTS FOR ENTHALPY EQUATION.                           SPD03830
  732 IF(LC.EQ.0) GO TO 999                                             SPD03840
      IF(L.LT.LC) GO TO 406                                             SPD03850
      LTABC=LTABC+1                                                     SPD03860
      IF(LTABC.EQ.LDELC) GO TO 407                                      SPD03870
      GO TO 408                                                         SPD03880
  407 LTABC=0                                                           SPD03890
  406 CALL SENTP                                                        SPD03900
  408 CONTINUE                                                          SPD03910
      DO 100 I=3,NP1                                                    SPD03920
  100 C(JH,I)=PD4*(BOMT3(I)*F(JH,I)+F(JH,I+1)*OMD(I)+F(JH,I-1)*OMD(I-1))SPD03930
     1+2.D0*SEN(I)                                                      SPD03940
C                                                                       SPD03950
C     ----------------------TEST 10 ------------------------------------SPD03960
      IF(ITEST(10).EQ.1) GO TO 733                                      SPD03970
      GO TO 734                                                         SPD03980
  733 WRITE(6,604)                                                      SPD03990
      WRITE(6,604)                                                      SPD04000
      WRITE(6,667)                                                      SPD04010
  667 FORMAT(1X,27HC COEFFICIENTS FOR ENTHALPY)                         SPD04020
      WRITE(6,626) (C(JH,I),I=3,NP1)                                    SPD04030
  626 FORMAT(1X,D14.5)                                                  SPD04040
C     ------------------------------------------------------------------SPD04050
C                                                                       SPD04060
C     COEFFICIENTS FOR THE ENTHALPY HAVE NOW BEEN CALCULATED.           SPD04070
  734 CALL CALC(JH)                                                     SPD04080
C                                                                       SPD04090
  999 L=L+1                                                             SPD04100
      LCOUNT=LCOUNT+1                                                   SPD04110
      IF(LCOUNT.EQ.LPRINT) GO TO 735                                    SPD04120
      GO TO 661                                                         SPD04130
  735 DO 800 J=1,JHM2                                                   SPD04140
      IF(.NOT.SPECIE(J)) GO TO 800                                      SPD04150
      CALL ENTRN(J,1)                                                   SPD04160
```

```
  800 CONTINUE                                                    SPD04170
      CALL ZCALC                                                  SPD04180
C     RESULTS PRINTED OUT.                                        SPD04190
      CALL OUTPUT(L,HEADNG,RUNID,X,INPUT,LMAX)                    SPD04200
      LCOUNT=0                                                    SPD04210
  661 IF(L.EQ.LMAX) WRITE(46) ((F(J,I),J=1,JH),I=1,NP3),PEI       SPD04220
      Q=L                                                         SPD04230
      DX=DXMIN*FAC2**Q                                            SPD04240
      X=X+DX                                                      SPD04250
      IF(L.EQ.LMAX) GO TO 1000                                    SPD04260
      GO TO 706                                                   SPD04270
 1000 CONTINUE                                                    SPD04280
      END                                                         SPD04290
```

```
@ELT,SIB ABCDE*FLAME(2).SWITCH,,,141371132312
      SUBROUTINE SWITCH(KRATE)                                          SWH00010
      LOGICAL G1,G2,G3                                                  SWH00020
      COMMON/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBODYA,JBODYB SWH00030
      COMMON/REACT/LA(30),LB(30),LRA(30),LRB(30)                        SWH00040
      COMMON/GSWITC /G1(30,15),G2(30,15),G3(30,15)                      SWH00050
      DATA G1/450*.FALSE./,G2/450*.FALSE./,G3/450*.FALSE./              SWH00060
      DO 15 J=1,JHM2                                                    SWH00070
      DO 15 K=1,KRATE                                                   SWH00080
      IF((J.EQ.LA(K)).AND.(J.EQ.LB(K)))  GO TO 10                       SWH00090
      IF(J.EQ.LA(K)) GO TO 11                                           SWH00100
      IF(J.EQ.LB(K)) GO TO 11                                           SWH00110
      GO TO 12                                                          SWH00120
   10 G3(K,J)=.TRUE.                                                    SWH00130
   11 G1(K,J)=.TRUE.                                                    SWH00140
   12 CONTINUE                                                          SWH00150
      IF((J.EQ.LRA(K)).AND.(J.EQ.LRB(K))) GO TO 13                      SWH00160
      IF(J.EQ.LRA(K)) GO TO 14                                          SWH00170
      IF(J.EQ.LRB(K)) GO TO 14                                          SWH00180
      GO TO 15                                                          SWH00190
   13 G3(K,J)=.TRUE.                                                    SWH00200
   14 G1(K,J)=.TRUE.                                                    SWH00210
      G2(K,J)=.TRUE.                                                    SWH00220
   15 CONTINUE                                                          SWH00230
      RETURN                                                            SWH00240
      END                                                               SWH00250
```

```
@ELT,SIB ABCDE*FLAME(2).DIFUSE,,,141343132312
      SUBRØUTINE DIFUSE(PRESS)
      REAL M(12)
      DØUBLE PRECISIØN TDIV,DA,DB,PRESS
      LØGICAL FLAG
      CØMMØN/JVALS/JB,JBM1,JBM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/TRANS4/DA(14,14),DB(14,14),TDIV(14,14)
      CØMMØN/FLAGS/FLAG(20)
      DIMENSIØN EK(12),S(12)
      DATA TDIV/196*.0D0/,DA/196*.0D0/,DB/196*.0D0/
      DATA M/1.008,16.,17.008,33.008,34.016,79.916,80.92,159.83
     * ,32.,18.016,2.016,28.02/,EK/51.6,555.,682.,298.,298.,342.
     * ,342.,403.5,88.,809.,38.,79.8 /,S/2.1,2.3,2.47,4.2,4.2
     * ,3.71,3.71,4.69,3.541,2.640,2.915,3.749/
      DØ 43 I=1,JBM2
      DØ 43 J=JRADP1,JBM1
      IF(I.GE.J) GØ TØ 43
      FMASS=(M(I)+M(J))/(M(I)*M(J))
      FMASS=SQRT(FMASS)
      EKIJ=SQRT(EK(I)*EK(J))
      EKA=EKIJ**.44
      EKB=EKIJ**.17
      S2=.5*(S(I)+S(J))
      S2=S2*S2
      TCUT=3.*EKIJ
      TDIV(I,J)=TCUT
      D=1.26E-7*FMASS/(S2*EKA)
      DA(I,J)=D
      IF(FLAG(3)) DA(I,J)=DA(I,J)/PRESS
      D=1.66E-7*FMASS/(S2*EKB)
      DB(I,J)=D
      IF(FLAG(3)) DB(I,J)=DB(I,J)/PRESS
   43 CØNTINUE
      RETURN
      END
```

```
@ELT,SIB ABCDE*FLAME(2).ØMEGA,,,141377132312
      SUBRØUTINE ØMEGA                                                  ØMG00010
      IMPLICIT REAL*8 (A-H,Ø-Z)                                         ØMG00020
      CØMMØN/IVALS/N,NP1,NP2,NP3                                        ØMG00030
      CØMMØN/ØMEGA1/NII,NEE,ØMR                                         ØMG00040
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)                     ØMG00050
      CØMMØN/ØMEGA3/BØM(43),BØMT3(43),ØMP(43),PEIØMD(43),PEIØM2(43)     ØMG00060
      CØMMØN/YCALC/PEI,Y(43),              YD(43),YAV(43),RPEI          ØMG00070
      NIIP1=NII+1                                                       ØMG00080
      NEEM1=NEE-1                                                       ØMG00090
      RØMR=1.DO/ØMR                                                     ØMG00100
      SUM=1.DO                                                          ØMG00110
      ØMD(2)=1.DO                                                       ØMG00120
      DØ 10 I=3,NII                                                     ØMG00130
      ØMD(I)=RØMR*ØMD(I-1)                                              ØMG00140
   10 SUM=SUM+ØMD(I)                                                    ØMG00150
      DØ 11 I=NIIP1,NEEM1                                               ØMG00160
      ØMD(I)=ØMD(NII)                                                   ØMG00170
   11 SUM=SUM+ØMD(I)                                                    ØMG00180
      DØ 12 I=NEE,NP1                                                   ØMG00190
      ØMD(I)=ØMR*ØMD(I-1)                                               ØMG00200
   12 SUM=SUM+ØMD(I)                                                    ØMG00210
      SUM=1.DO/SUM                                                      ØMG00220
      DØ 13 I=2,NP1                                                     ØMG00230
   13 ØMD(I)=ØMD(I)*SUM                                                 ØMG00240
      ØM(1)=.ODO                                                        ØMG00250
      ØM(2)=.ODO                                                        ØMG00260
      DØ 14 I=3,NP2                                                     ØMG00270
   14 ØM(I)=ØM(I-1)+ØMD(I-1)                                            ØMG00280
      ØM(NP3)=1.DO                                                      ØMG00290
      ØMI=.5DO*ØM(3)                                                    ØMG00300
```

```
      ØME=.5DO*(1.DO-ØM(NP1))                                   ØMG00310
      DØ 40 I=2,NP1                                             ØMG00320
      ØMD(I)=ØM(I+1)-ØM(I)                                      ØMG00330
      RØMD(I)=1.DO/ØMD(I)                                       ØMG00340
   40 ØMP(I)=ØM(I)+ØM(I+1)                                      ØMG00350
      DØ 50 I=3,NP1                                             ØMG00360
      BØM(I)=ØM(I+1)-ØM(I-1)                                    ØMG00370
   50 BØMT3(I)=3.DO*BØM(I)                                      ØMG00380
      RETURN                                                    ØMG00390
      END                                                       ØMG00400
```

```
@ELT,SIB ABCDE*FLAME(2).INITL/1,,,107650032113
      SUBRØUTINE INITL
      IMPLICIT REAL*8 (A-H,Ø-Z)
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)
      CØMMØN/YCALC/PEI,Y(43),            YD(43),YAV(43),RPEI
      CØMMØN/TRANS1/HREF(15),DCP(15),ECP(15),FCP(15),WT(15),ITER,ITER1
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/INIT/XI(15),XE(15),IPRFL(15),TCØLD,PRESS,YWIDTH
      CØMMØN/VAR1/F(15,43),FS(15,43)
      DIMENSIØN ET(15,43),Z(15),XN2(43)
      FNC(U)=U*U*U*(10.D0-15.D0*U+6.D0*U*U)
C     INITIAL PRØFILES FØR NØN-RADICALS.
C     XI(I) AND XE(I) ARE HØT AND CØLD SIDE MØLE FRACTIØNS ØF THE NØN-
C     RADICAL SPECIES.  WHEN PRFL(J)=0, WE HAVE A DECAY PRØFILE, AND
C     WHEN IPRFL(J)=1, A FØRMATIØN PRØFILE.
C
C     THIS SECTIØN CALCULATES MØLE FRACTIØN PRØFILES FØR ALL NØN-RADICAL
C     SPECIES EXCEPT ØNE.
C
      DØ 100 J=JRADP1,JHM1
      IF(IPRFL(J).EQ.1) GØ TØ 120
      DIFF=XE(J)-XI(J)
      DØ 130 I=2,NP2
      U=ØM(I)
  130 F(J,I)=XI(J)+DIFF*FNC(U)
      GØ TØ 100
  120 DIFF=XI(J)-XE(J)
      DØ 140 I=2,NP2
      U=1.D0-ØM(I)
  140 F(J,I)=XE(J)+DIFF*FNC(U)
  100 CØNTINUE
C
C     AT THIS PØINT, THE F(J,I) ARE MØLE FRACTIØNS.
C
C     THIS SECTIØN CALCULATES THE MØLE FRACTIØN PRØFILE FØR THE BUFFER
```

```
C     SPECIES.                                                          ITL00360
      DØ 150 I=2,NP2                                                    ITL00370
      XX=0.DO                                                           ITL00380
      DØ 160 J=JRADP1,JHM1                                              ITL00390
  160 XX=XX+F(J,I)                                                      ITL00400
  150 XN2(I)=1.DO-XX                                                    ITL00410
C     HERE, THE MØLE FRACTIØN PRØFILES ARE CØNVERTED TØ MØL/KG PRØFILES.ITL00420
      DØ 170 I=2,NP2                                                    ITL00430
      AVM=.ODO                                                          ITL00440
      DØ 180 J=JRADP1,JHM1                                              ITL00450
  180 AVM=AVM+F(J,I)*WT(J)                                              ITL00460
      RAVM=1.DO/ AVM                                                    ITL00470
      DØ 190 J=JRADP1,JHM1                                              ITL00480
  190 F(J,I)=F(J,I)*RAVM                                                ITL00490
  170 CØNTINUE                                                          ITL00500
C                                                                       ITL00510
C     INITIAL PRØFILES FØR RADICALS AND TRACE SPECIES.                  ITL00520
      CALL INPRFL(F,JRAD,N)                                             ITL00530
C                                                                       ITL00540
C     GIVEN THE TEMPERATURE ØN THE CØLD SIDE, WE CALCULATE THECØLD SIDE ITL00550
C     ENTHALPY.                                                         ITL00560
      T=TCØLD                                                           ITL00570
      T2=T*T                                                            ITL00580
      DØ 20 J=1,JHM1                                                    ITL00590
   20 ET(J,NP2)=HREF(J)+T*(DCP(J)+ECP(J)*T*.5D0+FCP(J)*T2*.333333D0)    ITL00600
      F(JH,NP2)=.0DO                                                    ITL00610
      DØ 30 J=JRADP1,JHM1                                               ITL00620
   30 F(JH,NP2)=F(JH,NP2)+WT(J)*F(J,NP2)*ET(J,NP2)                      ITL00630
C                                                                       ITL00640
C     THE ENTHALPY ØN THE HØT SIDE IS SET EQUAL TØ THAT ØN THE CØLD SIDEITL00650
      F(JH,2)=F(JH,NP2)                                                 ITL00660
C     NØW, THE HØT SIDE TEMPERATURE IS CALCULATED FRØM THE HØT SIDE     ITL00670
C     ENTHALPY AND CØMPØSITIØN. NØTE THAT Y(J) IS THE MASS FRACTIØN     ITL00680
C     ØN THE HØT SIDE.                                                  ITL00690
      A=.ODO                                                            ITL00700
```

```
      B=.0D0
      C2=.0D0
      D3=.0D0
      DØ 40 J=JRADP1,JHM1
      Z(J)=F(J,2)*WT(J)
      A=A+BREF(J)*Z(J)
      B=B+DCP(J)*Z(J)
      C2=C2+ECP(J)*Z(J)
   40 D3=D3+FCP(J)*Z(J)
      C=C2*.5D0
      D=D3*.333333D0
      T=(F(JH,2)-A)/B
      ICØUNT=0
   23 FAC=F(JH,2)-(A+T*(B+T*(C+D*T)))
      FAC=FAC/(B+T*(C2+D3*T))
      T=T+FAC
      ICØUNT=ICØUNT+1
      IF(ICØUNT.LT.ITER) GØ TØ 23
      THØT=T
C     AN S-SHAPED TEMPERATURE PRØFILE IS CALCULATED, AND FRØM THIS, THE
C     ENTHALPY PRØFILE.
      DØ 50 I=2,NP2
      U=1.D0-ØM(I)
   50 FS(JT,I)=(THØT-TCØLD)*FNC(U)+TCØLD
C     NØTE THAT CØNSTANT ENTHALPY IS USED AS A STARTING PRØFILE.
      DØ 70 I=3,NP2
   70 F(JH,I)=F(JH,2)
C
      RETURN
      END
```

```
@ELT,SIB ABCDE*FLAME(2).INPRFL,,,103730063613
      SUBROUTINE INPRFL(F,JRAD,N)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION F(15,43)
      NP2=N+2
      DO 10 I=2,NP2
      DO 20 J=1,4
   20 F(J,I)=.1D-4
      DO 21 J=5,8
   21 F(J,I)=1.D-50
   10 CONTINUE
      DO 22 J=1,8
   22 F(J,NP2)=1.D-50
      RETURN
      END
```

```
@ELT,SIB ABCDE*FLAME(2).TRANS/6,,,111236132413
      SUBRØUTINE TRANS(ICALC)
      IMPLICIT REAL*8 (A-H,Ø-Z)
      LØGICAL FLAG,TAGA,TAGB
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/VAR1/F(15,43),FS(15,43)
      CØMMØN/TRANS1/HREF(15),DCP(15),ECP(15),FCP(15),WT(15),ITER,ITER1
      CØMMØN/TRANS2/RMB(43),PREF(15,43),HDRM(15,43),TRCP(15,43)
     *,DR2M(15,43)
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)
      CØMMØN/INIT/XI(15),XE(15),IPRFL(15),TCØLD,PRESS,YWIDTH
      CØMMØN/TRANS4/DA(14,14),DB(14,14),TDIV(14,14)
      CØMMØN/FLAGS/FLAG(20)
      CØMMØN/CMINDX/LH,LØ,LØH,LHØ2,LH2Ø2,LX,LHX,LX2,LØ2,LH2Ø,LH2,LN2
      DIMENSIØN DD(14,14),CØN(14),ENT(15,43),SUM(14)
      DATA AH2/1.DO/,AØ2/1.DO/,AH2Ø/12.DO/
     * ,BH2/2.28DO/,BØ2/.8DO/,BH2Ø/14.8DO/
C     AVERAGE MØLECULAR WEIGHT AND MASS FRACTIØNS AT GRID PØINTS.
      DØ 10 I=2,NP2
      WTAV=.0DO
      A=.0DO
      B=.0DO
      C2=.0DO
      D3=.0DO
      DØ 14 J=1,JHM1
      Y=WT(J)*F(J,I)
      WTAV=WTAV+F(J,I)
      A=A+HREF(J)*Y
      B=B+DCP(J)*Y
      C2=C2+ECP(J)*Y
   14 D3=D3+FCP(J)*Y
      C=C2*.5DO
      D=D3*.333333DO
      FS(JMR,I)=WTAV
```

```
      FS(JM,I)=1.DO/WTAV
C
C     TEMPERATURE AT GRID PØINTS.
      T=(F(JH,I)-A)/B
      ICØUNT=0
   23 FAC=F(JH,I)-(A+T*(B+T*(C+D*T)))
      FAC=FAC/(B+T*(C2+D3*T))
      T=T+FAC
      ICØUNT=ICØUNT+1
      IF(ICØUNT.LT.ITER) GØ TØ 23
      FS(JT,I)=T
      FS(JTR,I)=1.DO/T
      FS(JBØDYA,I)=AH2*F(LH2,I)+AØ2*F(LØ2,I)+AH2Ø*F(LH2Ø,I)+F(LN2,I)
      FS(JBØDYB,I)=BH2*F(LH2,I)+BØ2*F(LØ2,I)+BH2Ø*F(LH2Ø,I)+F(LN2,I)
      IF(ICALC.EQ.0) GØ TØ 10
C
C     DENSITY AT GRID PØINTS.
      RHØ(I)=1.21867D4*FS(JM,I)*FS(JTR,I)
      IF(FLAG(3)) RHØ(I)=RHØ(I)*PRESS
      RH2(I)=RHØ(I)*RHØ(I)
C
C     EFFECTIVE DIFFUSIØN CØEFFICIENTS AT GRID PØINTS.
C
C     THIS SECTIØN CALCULATES PREF AT GRID PØINTS FØR Ø3 FLAME.
C     LEWIS NUMBER =.94.
      IF(.NØT.FLAG(7)) GØ TØ 211
      CPMIX=B+T*(C2+D3*T)
      PREF(JH,I)=2.2026D-3*DSQRT(T)
      PREF(JH,I)=PREF(JH,I)*RHØ(I)/CPMIX
      XLEWIS=.94D0
      FAC=XLEWIS*PREF(JH,I)
      DØ 212 J=1,JHM1
  212 PREF(J,I)=FAC
      GØ TØ 57
  211 CØNTINUE
```

```
C                                                                       TRN00700
C     THIS SECTION CALCULATES PREF AT GRID POINTS FOR H2-BR2 FLAME.     TRN00710
C     LEVACHEV AND KAGANOVA TRANSPORT PROPERTIES ARE USED.              TRN00720
      IF(.NOT.FLAG(8)) GO TO 214                                        TRN00730
      FAC=(T/TCOLD)**1.67                                               TRN00740
      DO 213 J=3,5                                                      TRN00750
  213 PREF(J,I)=.101D-4*FAC                                             TRN00760
      PREF(1,I)=.155D-4*FAC                                             TRN00770
      PREF(2,I)=1.05D-4*FAC                                             TRN00780
      DO 215 J=1,5                                                      TRN00790
  215 PREF(J,I)=PREF(J,I)*RH2(I)                                        TRN00800
      PREF(JH,I)=3.34944D-2*((T/TCOLD)**.67)                            TRN00810
      CPMIX=B+T*(C2+D3*T)                                               TRN00820
      PREF(JH,I)=PREF(JH,I)*RHO(I)/CPMIX                                TRN00830
      GO TO 57                                                          TRN00840
  214 CONTINUE                                                          TRN00850
C                                                                       TRN00860
C     IF FLAG(1)=T, CONSTANT TRANSPORT PROPERTIES ARE USED.             TRN00870
C     (VALUES OF PREF APPEAR IN A DATA STATEMENT IN SPALD.)             TRN00880
      IF(FLAG(1)) GO TO 10                                              TRN00890
      R2DM=RH2(I)*FS(JMR,I)                                             TRN00900
C                                                                       TRN00910
C     COMPOSITION DEPENDENT TRANSPORT PROPERTIES ARE CALCULATED IN THE  TRN00920
C     SECTION THAT FOLLOWS.                                             TRN00930
C                                                                       TRN00940
C     BINARY DIFFUSION COEFFICIENTS CALCULATED AT TEMPERATURE T.        TRN00950
      TAGA=.FALSE.                                                      TRN00960
      TAGB=.FALSE.                                                      TRN00970
      DO 37 KI=1,JHM2                                                   TRN00980
      DO 37 KJ=JRADP1,JHM1                                              TRN00990
      IF(KI.GE.KJ) GO TO 37                                             TRN01000
      IF((T.LT.TDIV(KI,KJ)).AND.(.NOT.TAGA)) GO TO 32                   TRN01010
      GO TO 33                                                          TRN01020
   32 TA=T**1.94                                                        TRN01030
                                                                        TRN01040
      TAGA=.TRUE.                                                       TRN01050
```

```
   33 IF(T.LT.TDIV(KI,KJ)) GØ TØ 34
      GØ TØ 35
   34 DD(KI,KJ)=DA(KI,KJ)*TA
      GØ TØ 30
   35 IF(TAGB) GØ TØ 36
      TB=T**1.67
      TAGB=.TRUE.
   36 DD(KI,KJ)=DB(KI,KJ)*TB
   30 CØNTINUE
   37 CØNTINUE
C
C     THE ABØVE DØ-LØØP CALCULATES THE BINARY DIFFUSIØN CØEFFICIENTS AT
C     THE TEMPERATURE T.  TAGA AND TAGB ARE USED SØ THAT THE EXPØNENTIA-
C     TIØN STEPS T**1.94 AND T**1.67 ARE CALCULATED ØNLY ØNCE AT EACH T.
C
C     IN THE NEXT SECTIØN, PREF(J,I) IS CALCULATED FØR THE FULL MØDEL.
      DØ 38 J=1,JHM1
      PREF(J,I)=.0D0
      DØ 39 K=JRADP1,JHM1
      IF(J.EQ.K) GØ TØ 39
      IF(J.GT.K) DD(J,K)=DD(K,J)
      PREF(J,I)=PREF(J,I)+F(K,I)/DD(J,K)
   39 CØNTINUE
      PREF(J,I)=R2DM/PREF(J,I)
      IF(J.GT.JRAD  ) PREF(J,I)=(1.D0-F(J,I)*FS(JM,I))*PREF(J,I)
   38 CØNTINUE
C
C     THIS SECTIØN CALCULATES THE THERMAL CØNDUCTIVITY FØR THE MIXTURE
C     ACCØRDING TØ LINDSAY AND BRØMLEY, IND. AND ENG. CHEM. VØL. 42,
C     P. 1508.  THE A(I,J) VALUES HAVE BEEN CALCULATED SEPARATELY FØR
C     THE FØUR MAJØR SPECIES, Ø2,H2Ø,H2,N2.
C
C     THE THERMAL CØNDUCTIVITIES FØR THE FØUR PURE SPECIES ARE
C     CALCULATED FIRST.
      T2=T*T
```

```
      T3=T*T2
      CØN(LØ2)=1.455D0+1.73669D-2*T-1.87901D-6*T2
      CØN(LH2Ø)=-2.51153D0+1.7143D-2*T+1.15567D-5*T2-3.15212D-9*T3
      CØN(LH2)=6.8173D0+.133568D0*T-3.89395D-5*T2+9.66224D-9*T3
      CØN(LN2)=9.85661D-2+2.2693D-2*T-9.4717D-6*T2+2.25781D-9*T3
      DØ 59 J=JRADP1,JHM1
   59 CØN(J)=4.1868D-3*CØN(J)
C
C     NEXT, THE A(I,J) CØEFFICIENTS ARE CALCULATED.
C     THE DD ARRAY IS USED FØR THESE QUANTITIES.
      DD(LØ2,LH2Ø)=135.4D0*FS(JTR,I)+.7654D0
      DD(LØ2,LH2)=.576D0
      DD(LØ2,LN2)=1.03D0
      DD(LH2Ø,LØ2)=.232322D0+1.2517D-3*T-6.94629D-7*T2+1.39243D-10*T3
      DD(LH2Ø,LH2)=.160533D0+5.95658D-4*T-3.17638D-7*T2+6.16877D-11*T3
      DD(LH2Ø,LN2)=.280067D0+1.08447D-3*T-4.78746D-7*T2+7.53136D-11*T3
      DD(LH2,LØ2)=1.98336D0+7.98D-5*T
      DD(LH2,LH2Ø)=309.29D0*FS(JTR,I)+1.4557D0
      DD(LH2,LN2)=2.09875D0+1.105D-4*T
      DD(LN2,LØ2)=.96429D0
      DD(LN2,LH2Ø)=138.68D0*FS(JTR,I)+.7244D0
      DD(LN2,LH2)=.5636D0
C
C     NØW, PREF(JH,I), WHICH IS THE THERMAL TRANSPØRT PARAMETER FØR THE
C     MIXTURE, IS CALCULATED.
C     FIRST, HØWEVER, THE MIXTURE HEAT CAPACITY AT THE GRID PØINTS IS
C     CALCULATED. ALSØ CALCULATED, IS THE SPECIES MØLAR ENTHALPY WHICH
C     WILL BE NEEDED FØR THE ENTHALPY SØURCE TERM.  THIS LATTER
C     CALCULATIØN WILL BE DØNE, HØWEVER, ØNLY IF THE ENTHALPY SØURCE
C     TERM IS TØ BE CALCULATED.  THIS ØPTIØN IS TRIGGERED BY FLAG(5)=T.
C
C     MIXTURE HEAT CAPACITY.
      CPMIX=B+T*(C2+D3*T)
C     MIXTURE THERMAL CØNDUCTIVITY AT GRID PØINTS.
      PREF(JH,I)=.0D0
```

```
      DØ 50 KI=JRADP1,JHM1
      SUM(KI)=.0D0
      DØ 52 KJ=JRADP1,JHM1
      IF(KI.EQ.KJ) GØ TØ 52
   52 SUM(KI)=SUM(KI)+DD(KI,KJ)*F(KJ,I)
      SUM(KI)=1.D0+SUM(KI)/F(KI,I)
   50 PREF(JH,I)=PREF(JH,I)+CØN(KI)/SUM(KI)
      PREF(JH,I)=PREF(JH,I)*RHØ(I)/CPMIX
      IF(FLAG(5)) GØ TØ 57
      GØ TØ 10
C     MØLAR EETHALPY AT GRID PØINTS.
   57 DØ 70 J=1,JHM1
      FAC=HREF(J)+T*(DCP(J)+ECP(J)*T*.5D0+FCP(J)*T2*.333333D0)
   70 ENT(J,I)=FAC*WT(J)
   10 CØNTINUE
C
      IF(ICALC.EQ.0) GØ TØ 300
C     VALUES AT CØNTRØL VØLUME BØUNDARIES.
      DØ 200 I=2,NP1
      RMB(I)=(FS(JMR,I)+FS(JMR,I+1))*.5D0
  200 PREF(JH,I)=(PREF(JH,I)+PREF(JH,I+1))*.5D0
      DØ 210 I=2,NP1
      DØ 210 J=1,JHM1
      PREF(J,I)=(PREF(J,I)+PREF(J,I+1))*.5D0
C     THE NEXT QUANTITIES APPEAR ØNLY IN THE ENTHALPY SØURCE TERM.
      IF(FLAG(5)) GØ TØ 58
      GØ TØ 210
   58 ENTLPY=.5*(ENT(J,I)+ENT(J,I+1))
      HDRM(J,I)=ENTLPY*PREF(J,I)*RMB(I)
      TRCP(J,I)=ENTLPY*PREF(JH,I)
      DR2M(J,I)=PREF(J,I)*RMB(I)
  210 CØNTINUE
  300 RETURN
      END
```

```
@ELT,SIB ABCDE*FLAME(2).RATCN/3,,,132060133512
      SUBRØUTINE RATCN
      IMPLICIT REAL*8 (A-H,Ø-Z)
      LØGICAL BØD1,BØD2,REACT,BFLAG,EFLAG,FFLAG,RFLAG
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/VAR1/F(15,43),FS(15,43)
      CØMMØN/RTPARM/FRQ(30),BETA(30),EACT(30)
     *,BØD1(30),BØD2(30),REACT(30)
      CØMMØN/RATC/RATE(30,43),RATER(30,43),KRATE,EQA(30),EQB(30),EQC(30)
      CØMMØN/FLAGR/BFLAG(30),EFLAG(30),FFLAG(30),RFLAG(30)
      DIMENSIØN ECØN(43),RGT(43)
      DØ 68 I=2,NP2
   68 RGT(I)=1.21867D4*FS(JTR,I)
      DØ 40 K=1,KRATE
      IF(.NØT.FFLAG(K)) GØ TØ 50
      IF((.NØT.BFLAG(K)).AND.(.NØT.EFLAG(K))) GØ TØ 10
      GØ TØ 12
   10 DØ 11 I=2,NP2
   11 RATE(K,I)=FRQ(K)
      GØ TØ 40
   12 IF(.NØT.BFLAG(K)) GØ TØ 13
      GØ TØ 15
   13 DØ 14 I=2,NP2
      E=-EACT(K)*FS(JTR,I)
   14 RATE(K,I)=FRQ(K)*DEXP(E)
      GØ TØ 40
   15 IF(.NØT.EFLAG(K)) GØ TØ 16
      GØ TØ 17
   16 DØ 18 I=2,NP2
   18 RATE(K,I)=FRQ(K)*(FS(JT,I)**BETA(K))
      GØ TØ 40
   17 DØ 19 I=2,NP2
      E=-EACT(K)*FS(JTR,I)
   19 RATE(K,I)=FRQ(K)*(FS(JT,I)**BETA(K))*DEXP(E)
```

```
      GØ TØ 40                                                          RCN00350
   50 DØ 67 I=2,NP2                                                     RCN00360
   67 RATE(K,I)=.0D0                                                    RCN00370
   40 CØNTINUE                                                          RCN00380
      DØ 60 K=1,KRATE                                                   RCN00390
      IF(.NØT.RFLAG(K)) GØ TØ 61                                        RCN00400
      DØ 62 I=2,NP2                                                     RCN00410
      ECØN(I)=-EQA(K)*FS(JTR,I)-EQB(K)-EQC(K)*FS(JT,I)                  RCN00420
   62 ECØN(I)=DEXP(ECØN(I))                                             RCN00430
      IF(BØD1(K)) GØ TØ 128                                             RCN00440
      GØ TØ 124                                                         RCN00450
  128 IF(BØD2(K)) GØ TØ 126                                             RCN00460
      GØ TØ 122                                                         RCN00470
  124 DØ 63 I=2,NP2                                                     RCN00480
   63 RATER(K,I)=RATE(K,I)/ECØN(I)                                      RCN00490
      GØ TØ 60                                                          RCN00500
  122 DØ 64 I=2,NP2                                                     RCN00510
   64 RATER(K,I)=RATE(K,I)*RGT(I)/ECØN(I)                               RCN00520
      GØ TØ 60                                                          RCN00530
  126 DØ 65 I=2,NP2                                                     RCN00540
   65 RATER(K,I)=RATE(K,I)/(ECØN(I)*RGT(I))                             RCN00550
      GØ TØ 60                                                          RCN00560
   61 DØ 66 I=2,NP2                                                     RCN00570
   66 RATER(K,I)=.0D0                                                   RCN00580
   60 CØNTINUE                                                          RCN00590
      RETURN                                                            RCN00600
      END                                                               RCN00610
```

```
@ELT,SI B ABCDE*FLAME(2).ENTRN/3,,,141333132312
      SUBRØUTINE ENTRN(J,IFØW)                                          ENT00010
      IMPLICIT REAL*8 (A-H,Ø-Z)                                         ENT00020
      REAL ALPHA                                                        ENT00030
      CØMMØN/IVALS/N,NP1,NP2,NP3                                        ENT00040
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB ENT00050
      CØMMØN/ØMEGA1/NII,NEE,ØMR                                         ENT00060
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)                     ENT00070
      CØMMØN/TRANS1/HREF(15),DCP(15),ECP(15),FCP(15),WT(15),ITER,ITER1  ENT00080
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)                              ENT00090
      CØMMØN/ØMEGA3/BØM(43),BØMT3(43),ØMP(43),PEIØMD(43),PEIØM2(43)     ENT00100
      CØMMØN/VAR1/F(15,43),FS(15,43)                                    ENT00110
      CØMMØN/ENTR/VEL(15),RMI,RME,DX,ALPHA                              ENT00120
      CØMMØN/YCALC/PEI,Y(43),YD(43),YAV(43),RPEI                        ENT00130
      CØMMØN/MASPRØ/R(15,43)                                            ENT00140
   12 SUM=.0D0                                                          ENT00150
      DØ 30 I=3,NP1                                                     ENT00160
   30 SUM=SUM+R(J,I)*BØMT3(I)+R(J,I+1)*ØMD(I)+R(J,I-1)*ØMD(I-1)         ENT00170
      SUM=.125D0*PEI*DABS(SUM)                                          ENT00180
      H=F(J,2)                                                          ENT00190
      HH=F(J,NII)                                                       ENT00200
      C=F(J,NP2)                                                        ENT00210
      CC=F(J,NEE)                                                       ENT00220
      FAC=1.D0/DABS(H-C)                                                ENT00230
      SUM=-SUM*FAC                                                      ENT00240
      VEL(J)=SUM/RHØ(NP2)                                               ENT00250
C                                                                       ENT00260
      IF(IFØW.EQ.1) GØ TØ 500                                           ENT00270
      RMI=(H-HH)*FAC*100.D0                                             ENT00280
      RMI=DABS(RMI)                                                     ENT00290
      RMI=RMI**ALPHA                                                    ENT00300
      RMI=2.D0-RMI                                                      ENT00310
      RMI=SUM*RMI                                                       ENT00320
      RME=(CC-C)*FAC*100.D0                                             ENT00330
      RME=DABS(RME)                                                     ENT00340
```

```
      RME=RME**ALPHA
      RME=SUM*RME
      PEI=PEI+(RMI-RME)*DX
  500 RETURN
      END
```

```
@ELT,SIH ABCDE*FLAME(2).SCHM1/1,,,14137313 2312
      SUBRØUTINE SCHM1                                                  SC100010
      IMPLICIT REAL*8 (A-H,Ø-Z)                                         SC100020
      CØMMØN/IVALS/N,NP1,NP2,NP3                                        SC100030
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB SC100040
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)                     SC100050
      CØMMØN/TRANS2/RMB(43),PREF(15,43),HDRM(15,43),TRCP(15,43)         SC100060
     *,DR2M(15,43)                                                      SC100070
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)                              SC100080
      CØMMØN/VAR1/F(15,43),FS(15,43)                                    SC100090
      CØMMØN/YCALC/PEI,Y(43),YD(43),YAV(43),RPEI                        SC100100
      CØMMØN/SCM1/SC1(15,43)                                            SC100110
      DIMENSIØN BF(15,43)                                               SC100120
      DØ 10 I=2,NP2                                                     SC100130
      DØ 10 J=1,JHM2                                                    SC100140
      BF(J,I)=(F(J,I)+F(J,I+1))*.5D0                                    SC100150
   10 CØNTINUE                                                          SC100160
      DØ 20 I=3,NP1                                                     SC100170
      DØ 20 J=1,JHM2                                                    SC100180
      FAC=PREF(J,I)      *RMB(I)*BF(J,I)*(FS(JM,I+1)-FS(JM,I))*RØMD(I)  SC100190
      FAC=FAC-PREF(J,I-1)       *RMB(I-1)*BF(J,I-1)*(FS(JM,I)           SC100200
     1-FS(JM,I-1))*RØMD(I-1)                                            SC100210
      SC1(J,I)=FAC*RPEI                                                 SC100220
   20 CØNTINUE                                                          SC100230
      RETURN                                                            SC100240
      END                                                               SC100250
```

```
@ELT,SIH ABCDE*FLAME(2).SCHM2/7,,,141405132312
      SUBRØUTINE SCHM2(J)
      IMPLICIT REAL*8 (A-H,Ø-Z)
      LØGICAL G1,G2,G3,BØD1,BØD2
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)
      CØMMØN/VAR1/F(15,43),FS(15,43)
      CØMMØN/ØMEGA3/BØM(43),BØMT3(43),ØMP(43),PEIØMD(43),PEIØM2(43)
      CØMMØN/RATC/RATE(30,43),RATER(30,43),KRATE,EQA(30),EQB(30),EQC(30)
      CØMMØN/SCM2/SU(15,43),SD(15,43)
      CØMMØN/MASPRØ/R(15,43)
      CØMMØN/GSWITC /G1(30,15),G2(30,15),G3(30,15)
      CØMMØN/REACT/LA(30),LB(30),LRA(30),LRB(30)
      CØMMØN/RTPARM/FRQ(30),BETA(30),EACT(30)
     * ,BØD1(30),BØD2(30),REACT(30)
      DIMENSIØN FØR(43),REV(43)
      DØ 60 I=2,NP2
      SU(J,I)=.0D0
   60 SD(J,I)=.0D0
      DØ 18 K=1,KRATE
      IF(.NØT.G1(K,J)) GØ TØ 18
      LAKKK=LA(K)
      LBKKK=LB(K)
      LRAKKK=LRA(K)
      LRBKKK=LRB(K)
      IF(K.EQ.5) GØ TØ 47
      IF(K.EQ.10) GØ TØ 48
      IF(BØD1(K)) GØ TØ 41
      DØ 50 I=2,NP2
      FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)
   50 REV(I)=RATER(K,I)*F(LRAKKK,I)*F(LRBKKK,I)
      GØ TØ 40
   41 IF(BØD2(K)) GØ TØ 42
      DØ 51 I=2,NP2
```

```
      FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)*FS(JMR,I)*RHØ(I)       SC200350
   51 REV(I)=RATER(K,I)*F(LRAKKK,I)*FS(JMR,I)                       SC200360
      GØ TØ 40                                                      SC200370
   42 DØ 53 I=2,NP2                                                 SC200380
      FØR(I)=RATE(K,I)*F(LAKKK,I)*FS(JMR,I)                         SC200390
   53 REV(I)=RATER(K,I)*F(LRAKKK,I)*F(LRBKKK,I)*FS(JMR,I)*RHØ(I)    SC200400
      GØ TØ 40                                                      SC200410
   47 DØ 54 I=2,NP2                                                 SC200420
      FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)*FS(JBØDYA,I)*RHØ(I)    SC200430
   54 REV(I)=RATER(K,I)*F(LRAKKK,I)*FS(JBØDYA,I)                    SC200440
      GØ TØ 40                                                      SC200450
   48 DØ 55 I=2,NP2                                                 SC200460
      FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)*FS(JBØDYB,I)*RHØ(I)    SC200470
   55 REV(I)=RATER(K,I)*F(LRAKKK,I)*FS(JBØDYB,I)                    SC200480
   40 CØNTINUE                                                      SC200490
      IF(G3(K,J)) GØ TØ 19                                          SC200500
      IF(G2(K,J)) GØ TØ 20                                          SC200510
      DØ 56 I=2,NP2                                                 SC200520
      SU(J,I)=SU(J,I)+REV(I)                                        SC200530
   56 SD(J,I)=SD(J,I)-FØR(I)                                        SC200540
      GØ TØ 18                                                      SC200550
   20 DØ 57 I=2,NP2                                                 SC200560
      SU(J,I)=SU(J,I)+FØR(I)                                        SC200570
   57 SD(J,I)=SD(J,I)-REV(I)                                        SC200580
      GØ TØ 18                                                      SC200590
   19 IF(G2(K,J)) GØ TØ 21                                          SC200600
      DØ 58 I=2,NP2                                                 SC200610
      SU(J,I)=SU(J,I)+2.DO*(FØR(I)+REV(I))                          SC200620
   58 SD(J,I)=SD(J,I)-4.DO*FØR(I)                                   SC200630
      GØ TØ 18                                                      SC200640
   21 DØ 59 I=2,NP2                                                 SC200650
      SU(J,I)=SU(J,I)+2.DO*(FØR(I)+REV(I))                          SC200660
   59 SD(J,I)=SD(J,I)-4.DO*REV(I)                                   SC200670
   18 CØNTINUE                                                      SC200680
      DØ 30 I=2,NP2                                                 SC200690
```

```
      SU(J,I)=SU(J,I)*RHØ(I)                                    SC200700
      SD(J,I)=SD(J,I)*RHØ(I)                                    SC200710
      R(J,I)=SU(J,I)+SD(J,I)                                    SC200720
   30 SD(J,I)=SD(J,I)/F(J,I)                                    SC200730
      DØ 31 I=3,NP1                                             SC200740
      SU(J,I)=SU(J,I)*PEIØM2(I)                                 SC200750
   31 SD(J,I)=SD(J,I)*PEIØM2(I)                                 SC200760
      RETURN                                                    SC200770
      END                                                       SC200780
```

```
@ELT,SIH ABCDE*FLAME(2).CALC/1,,,137751133112
      SUBRØUTINE CALC(J)
      IMPLICIT REAL*8 (A-H,Ø-Z)
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/VAR1/F(15,43),FS(15,43)
      CØMMØN/CØEF/A(15,43),B(15,43),C(15,43),D(15,43)
      A(J,3)=A(J,3)/D(J,3)
      B(J,3)=(B(J,3)*F(J,2)+C(J,3))/D(J,3)
      DØ 20 I=4,NP1
      T=D(J,I)-B(J,I)*A(J,I-1)
      A(J,I)=A(J,I)/T
   20 B(J,I)=(B(J,I)*B(J,I-1)+C(J,I))/T
      DØ 30 IDASH=3,NP1
      I=N+4-IDASH
      F(J,I)=A(J,I)*F(J,I+1)+B(J,I)
      IF(J.EQ.JH) GØ TØ 30
      IF(F(J,I).LT..0D0) F(J,I)=1.D-30
   30 CØNTINUE
      RETURN
      END
```

```
@ELT,SIB ABCDE*FLAME(2).SENTP/1,,,141375132312
      SUBRØUTINE SENTP
      IMPLICIT REAL*8 (A-H,Ø-Z)
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)
      CØMMØN/TRANS2/RMB(43),PREF(15,43),HDRM(15,43),TRCP(15,43)
     *,DR2M(15,43)
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)
      CØMMØN/VAR1/F(15,43),FS(15,43)
      CØMMØN/YCALC/PEI,Y(43),              YD(43),YAV(43),RPEI
      CØMMØN/SENT /SEN(43)
      DØ 20 I=3,NP1
      SEN(I)=.0D0
      DØ 10 J=1,JHM1
      FAC=HDRM(J,I)*(F(J,I+1)*FS(JM,I+1)-F(J,I)*FS(JM,I))*RØMD(I)
      FAC=FAC-HDRM(J,I-1)*(F(J,I)*FS(JM,I)-F(J,I-1)*FS(JM,I-1))
     1*RØMD(I-1)
      FAC=FAC-TRCP(J,I)*(F(J,I+1)-F(J,I))*RØMD(I)
      FAC=FAC+TRCP(J,I-1)*(F(J,I)-F(J,I-1))*RØMD(I-1)
      SEN(I)=SEN(I)+FAC
   10 CØNTINUE
      SEN(I)=SEN(I)*RPEI
   20 CØNTINUE
      RETURN
      END
```

```
@ELT,SIB ABCDE*FLAME(2).ZCALC/1,,,141401132312
      SUBRØUTINE ZCALC
      IMPLICIT REAL*8 (A-H,Ø-Z)
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)
      CØMMØN/YCALC/PEI,Y(43),          YD(43),YAV(43),RPEI
      DØ 50 I=2,NP1
   50 RU(I)=(RHØ(I)+RHØ(I+1))*.5D0
      Y(1.)=.0D0
      Y(2)=.0D0
      DØ 20 I=3,NP2
   20 Y(I)=Y(I-1)+PEI*ØMD(I-1)/RU(I-1)
      Y(NP3)=Y(NP2)
      DØ 30 I=2,NP1
   30 YD(I)=Y(I+1)-Y(I)
      DØ 40 I=3,NP1
   40 YAV(I)=(Y(I+1)-Y(I-1))*.5D0
      RETURN
      END
```

```
@ELT,SIH ABCDE*FLAME(2).ØUTPUT,,,142032031413
      SUBRØUTINE ØUTPUT(L,HEADNG,RUNID,X,INPUT,LMAX)
      IMPLICIT REAL*8 (A-H,Ø-Z)
      REAL V(30,43),W(30,43), XMAX(30),YMAX(30),Z(60)
      REAL ZTAB,TAB,PUT(15),HV(30,43),HW(30,43),SFAC
      LØGICAL SKIP(60),FLAG
      DIMENSIØN IX(30),IY(30)
      CØMMØN/IVALS/N,NP1,NP2,NP3
      CØMMØN/JVALS/JH,JHM1,JHM2,JRAD,JRADP1,JM,JMR,JT,JTR,JBØDYA,JBØDYB
      CØMMØN/ØMEGA1/NII,NEE,ØMR
      CØMMØN/ØMEGA2/ØM(43),ØMD(43),ØMI,ØME,RØMD(43)
      CØMMØN/INIT/XI(15),XE(15),IPRFL(15),TCØLD,PRESS,YWIDTH
      CØMMØN/TRANS1/HREF(15),DCP(15),ECP(15),FCP(15),WT(15),ITER,ITER1
      CØMMØN/TRANS2/RMB(43),PREF(15,43),HDRM(15,43),TRCP(15,43)
     *,DR2M(15,43)
      CØMMØN/TRANS3/RHØ(43),RU(43),RH2(43)
      CØMMØN/VAR1/F(15,43),FS(15,43)
      CØMMØN/YCALC/PEI,Y(43),YD(43),YAV(43),RPEI
      CØMMØN/ENTR/VEL(15),RMI,RME,DX,ALPHA
      CØMMØN/MASPRØ/R(15,43)
      CØMMØN/FLAGS/FLAG(20)
      REAL FFØR(43),RREV(43)
      LØGICAL BØD1,BØD2
      CØMMØN/RTPARM/FRQ(30),BETA(30),EACT(30)
     * ,BØD1(30),BØD2(30),REACT(30)
      CØMMØN/RATC/RATE(30,43),RATER(30,43),KRATE,EQA(30),EQB(30),EQC(30)
      CØMMØN/REACT/LA(30),LB(30),LRA(30),LRB(30)
      REAL HEADNG(126),ØUT(14),ØUTH,ØUTT,ØUTM,RUNID(12)
      DIMENSIØN FØR(43),REV(43)
      DATA ØUT/14*.0/,SKIP/60*.FALSE./
   69 FØRMAT(1H )
      WRITE(6,10) L,DX,X
   10 FØRMAT(1H1,'NUMBER ØF STEPS TAKEN=',I3,3X,'STEP SIZE FØR LAST STEP
     *=',D12.5,3X,'ELAPSED TIME=',D12.5 )
      WRITE(6,69)
```

```
      WRITE(6,11) (HEADNG(M),M=1,126)                                   ØUT00350
   11 FØRMAT(3X,126A1,1X,'I')                                           ØUT00360
      DØ 12 I=2,NP2                                                     ØUT00370
      DØ 13 J=1,JHM1                                                    ØUT00380
   13 ØUT(J)=F(J,I)                                                     ØUT00390
      ØUTH=F(JH,I)                                                      ØUT00400
      ØUTT=FS(JT,I)                                                     ØUT00410
      ØUTM=FS(JM,I)                                                     ØUT00420
      WRITE(6,15) ((ØUT(J),J=1,12),ØUTH,ØUTT,I)                         ØUT00430
   15 FØRMAT(2X,12E9.5,E10.5,E9.5,I3)                                   ØUT00440
   12 CØNTINUE                                                          ØUT00450
      WRITE(6,69)                                                       ØUT00460
      WRITE(6,25)                                                       ØUT00470
   25 FØRMAT( 50X,'FLAME VELØCITIES ')                                  ØUT00480
      DØ 26 J=1,JHM1                                                    ØUT00490
   26 ØUT(J)=VEL(J)                                                     ØUT00500
      WRITE(6,27) (ØUT(J),J=1,JHM2)                                     ØUT00510
   27 FØRMAT(3X,14E9.4)                                                 ØUT00520
      WRITE(6,69)                                                       ØUT00530
      WRITE(6,28) PEI,RMI,RME                                           ØUT00540
   28 FØRMAT(' GRID WIDTH; PEI=',D12.5,5X,'ENTRAINMENT RATES:',         ØUT00550
     *1X,'RMI=',D12.5,2X,'RME=',D12.5)                                  ØUT00560
      IF(L.EQ.LMAX) GØ TØ 16                                            ØUT00570
      GØ TØ 99                                                          ØUT00580
   16 WRITE(6,17)                                                       ØUT00590
   17 FØRMAT(1H1,50X,'PARAMETERS FØR THIS RUN' )                        ØUT00600
      WRITE(6,69)                                                       ØUT00610
      WRITE(6,18) N,NII,NEE,ØMR                                         ØUT00620
   18 FØRMAT(1X,'GRID STRUCTURE:',3X,'N=',I2,2X,'NII=',I2,2X,'NEE=',    ØUT00630
     *I2,2X,'ØMR=',D8.2)                                                ØUT00640
      WRITE(6,19) JH,JRAD,JM,JMR,JT,JTR,JBØDYA,JBØDYB                   ØUT00650
   19 FØRMAT(1X,'J INDICES:',3X,'JH=',I2,2X,'JRAD=',I2,2X,'JM=',I2,     ØUT00660
     *2X,'JMR=',I2,2X,'JT=',I2,2X,'JTR=',I2,2X,'JBØDY=',I2,2X,'JBØDYB=',ØUT00670
     *I2)                                                               ØUT00680
      WRITE(6,20) TCØLD,PRESS,INPUT                                     ØUT00690
```

```
20 FØRMAT(1X,'TEMPERATURE AT CØLD SIDE=',D12.5,3X,'PRESSURE=',
  *D12.5,3X,'INPUT=',I1/1X,'(IF INPUT=0, PRØFILES WERE CALCULATED BY
  *INITL; IF INPUT=1, THEY WERE TAKEN FRØM 45. THE ØUTPUT WENT INTØ 4
  *6.)')
   WRITE(6,69)
C  FØR(I) ARRAY IS USED FØR Q(I), THE HEAT RELEASE RATE.
C  FIRST, THE PRØDUCTIØN RATE ØF THE JHM2 SPECIES IS CALCULATED
C  IN THE CASE THAT THE CØNSTANT N2 APPRØXIMATIØN IS BEING USED.
   IF(FLAG(2)) GØ TØ 90
   JHM3=JH-3
   DØ 91 I=2,NP2
   SUM=.0D0
   DØ 92 J=1,JHM3
92 SUM=SUM+R(J,I)*WT(J)
91 R(JHM2,I)=-SUM/WT(JHM2)
90 DØ 70 I=2,NP2
   FØR(I)=.0D0
   T=FS(JT,I)
   T2=T*T
C  NØTE THAT R(J,I)=0 FØR THE JHM1 SPECIES IF IT IS A BUFFER.
   DØ 71 J=1,JHM1
   FAC=HREF(J)+T*(DCP(J)+ECP(J)*T*.5D0+FCP(J)*T2*.333333D0)
   PUT(J)=FAC*WT(J)
71 FØR(I)=FØR(I)+R(J,I)*FAC*WT(J)
70 CØNTINUE
   WRITE(6,69)
   WRITE(6,23)
23 FØRMAT(  1X,'SPACE VARIABLES')
   WRITE(6,21)
21 FØRMAT(2X,'I',7X,'ØMEGA',9X,'Y',9X,'DENSITY',4X,'AV MØL WT'
  * ,4X,'HEAT REL.')
   WRITE(6,22) (I,ØM(I),Y(I),RHØ(I),FS(JM,I),FØR(I),I=2,NP2)
22 FØRMAT(1X,I3,2X,5D12.5)
   WRITE(6,69)
   WRITE(6,69)
```

```
      WRITE(6,24) (RUNID(M),M=1,12)                                     ØUT01050
   24 FØRMAT(  1X,'RUN IDENTIFICATIØN:',3X,12A6)                        ØUT01060
      WRITE(6,29)                                                       ØUT01070
   29 FØRMAT(1H1,10X,'MØLE PRØDUCTIØN RATES ØF THE VARIØUS CHEMICAL SPECØUT01080
     *IES' )                                                            ØUT01090
      WRITE(6,69)                                                       ØUT01100
      WRITE(6,11) (HEADNG(M),M=1,118)                                   ØUT01110
      DØ 30 I=2,NP2                                                     ØUT01120
      DØ 31 J=1,JHM1                                                    ØUT01130
   31 ØUT(J)=R(J,I)                                                     ØUT01140
      DØ 32 J=JH,14                                                     ØUT01150
   32 ØUT(J)=.0                                                         ØUT01160
      WRITE(6,33) ((ØUT(J),J=1,14),I)                                   ØUT01170
   33 FØRMAT(3X,14E9.4,I3)                                              ØUT01180
   30 CØNTINUE                                                          ØUT01190
      WRITE(6,34)                                                       ØUT01200
   34 FØRMAT(1H1,'TRANSPØRT PARAMETERS; I.E., THE ARRAY PREF')          ØUT01210
      WRITE(6,69)                                                       ØUT01220
      WRITE(6,11) (HEADNG(M),M=1,118)                                   ØUT01230
      JHP1=JH+1                                                         ØUT01240
      DØ 35 I=2,NP2                                                     ØUT01250
      DØ 36 J=1,JH                                                      ØUT01260
   36 ØUT(J)=PREF(J,I)                                                  ØUT01270
      DØ 37 J=JHP1,14                                                   ØUT01280
   37 ØUT(J)=.0                                                         ØUT01290
      WRITE(6,33) ((ØUT(J),J=1,14),I)                                   ØUT01300
   35 CØNTINUE                                                          ØUT01310
      IF(.NØT.FLAG(9)) GØ TØ 100                                        ØUT01320
      ICØUNT=0                                                          ØUT01330
      ISTART=4                                                          ØUT01340
  104 ISTØP=ISTART+15                                                   ØUT01350
      WRITE(6,101)                                                      ØUT01360
  101 FØRMAT(1H1,'VALUES ØF RATE CØNSTANTS AT EACH PØINT IN THE FLAME') ØUT01370
      WRITE(6,69)                                                       ØUT01380
      WRITE(6,44) (I,I=ISTART,ISTØP)                                    ØUT01390
```

```
C     NØTE THAT FFØR AND RREV ARRAYS ARE USED TØ HØLD RATE AND RATER.   ØUT01400
      DØ 102 K=1,KRATE                                                  ØUT01410
      DØ 103 I=ISTART,ISTØP                                             ØUT01420
      FFØR(I)=RATE(K,I)                                                 ØUT01430
  103 RREV(I)=RATER(K,I)                                                ØUT01440
      WRITE(6,43) (K,(FFØR(I),I=ISTART,ISTØP))                          ØUT01450
      WRITE(6,43) (K,(RREV(I),I=ISTART,ISTØP))                          ØUT01460
  102 CØNTINUE                                                          ØUT01470
      ICØUNT=ICØUNT+1                                                   ØUT01480
      IF(ICØUNT.EQ.2) GØ TØ 100                                         ØUT01490
      ISTART=20                                                         ØUT01500
      GØ TØ 104                                                         ØUT01510
  100 CØNTINUE                                                          ØUT01520
      IF(.NØT.FLAG(10)) GØ TØ 45                                        ØUT01530
      ICØUNT=0                                                          ØUT01540
      ISTART=4                                                          ØUT01550
   46 ISTØP=ISTART+15                                                   ØUT01560
      WRITE(6,38)                                                       ØUT01570
   38 FØRMAT(1H1,'RATES ØF EACH REACTIØN AT EACH PØINT IN THE FLAME')   ØUT01580
      WRITE(6,69)                                                       ØUT01590
      WRITE(6,44) (I,I=ISTART,ISTØP)                                    ØUT01600
   44 FØRMAT(1X,16(3X,I2,3X))                                           ØUT01610
      DØ 39 K=1,KRATE                                                   ØUT01620
      LAKKK=LA(K)                                                       ØUT01630
      LBKKK=LB(K)                                                       ØUT01640
      LRAKKK=LRA(K)                                                     ØUT01650
      LRBKKK=LRB(K)                                                     ØUT01660
      DØ 50 I=ISTART,ISTØP                                              ØUT01670
      IF(K.EQ.5) GØ TØ 47                                               ØUT01680
      IF(K.EQ.10) GØ TØ 48                                              ØUT01690
      IF(BØD1(K)) GØ TØ 41                                              ØUT01700
      FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)*RHØ(I)                     ØUT01710
      REV(I)=RATER(K,I)*F(LRAKKK,I)*F(LRBKKK,I)*RHØ(I)                  ØUT01720
      SFAC=PUT(LRAKKK)+PUT(LRBKKK)-PUT(LAKKK)-PUT(LBKKK)                ØUT01730
      FFØR(I)=FØR(I)                                                    ØUT01740
```

```
      RREV(I)=REV(I)                                                    ØUT01750
      HV(K,I)=SFAC*FFØR(I)                                              ØUT01760
      HW(K,I)=-SFAC*RREV(I)                                             ØUT01770
      GØ TØ 40                                                          ØUT01780
   41 IF(BØD2(K)) GØ TØ 42                                              ØUT01790
      FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)*FS(JMR,I)*RH2(I)           ØUT01800
      REV(I)=RATER(K,I)*F(LRAKKK,I)*FS(JMR,I)*RHØ(I)                    ØUT01810
      SFAC=PUT(LRAKKK)-PUT(LAKKK)-PUT(LBKKK)                            ØUT01820
      FFØR(I)=FØR(I)                                                    ØUT01830
      RREV(I)=REV(I)                                                    ØUT01840
      HV(K,I)=SFAC*FFØR(I)                                              ØUT01850
      HW(K,I)=-SFAC*RREV(I)                                             ØUT01860
      GØ TØ 40                                                          ØUT01870
   42 FØR(I)=RATE(K,I)*F(LAKKK,I)*FS(JMR,I)*RHØ(I)                      ØUT01880
      REV(I)=RATER(K,I)*F(LRAKKK,I)*F(LRBKKK,I)*FS(JMR,I)*RH2(I)        ØUT01890
      SFAC=PUT(LRAKKK)+PUT(LRBKKK)-PUT(LAKKK)                           ØUT01900
      FFØR(I)=FØR(I)                                                    ØUT01910
      RREV(I)=REV(I)                                                    ØUT01920
      HV(K,I)=SFAC*FFØR(I)                                              ØUT01930
      HW(K,I)=-SFAC*RREV(I)                                             ØUT01940
      GØ TØ 40                                                          ØUT01950
   47 FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)*FS(JBØDYA,I)*RH2(I)        ØUT01960
      REV(I)=RATER(K,I)*F(LRAKKK,I)*FS(JBØDYA,I)*RHØ(I)                 ØUT01970
      SFAC=PUT(LRAKKK)-PUT(LAKKK)-PUT(LBKKK)                            ØUT01980
      FFØR(I)=FØR(I)                                                    ØUT01990
      RREV(I)=REV(I)                                                    ØUT02000
      HV(K,I)=SFAC*FFØR(I)                                              ØUT02010
      HW(K,I)=-SFAC*RREV(I)                                             ØUT02020
      GØ TØ 40                                                          ØUT02030
   48 FØR(I)=RATE(K,I)*F(LAKKK,I)*F(LBKKK,I)*FS(JBØDYB,I)*RH2(I)        ØUT02040
      REV(I)=RATER(K,I)*F(LRAKKK,I)*FS(JBØDYB,I)*RHØ(I)                 ØUT02050
      SFAC=PUT(LRAKKK)-PUT(LAKKK)-PUT(LBKKK)                            ØUT02060
      FFØR(I)=FØR(I)                                                    ØUT02070
      RREV(I)=REV(I)                                                    ØUT02080
      HV(K,I)=SFAC*FFØR(I)                                              ØUT02090
```

```
      BW(K,I)=-SFAC*RREV(I)                                             ØUT02100
   40 CØNTINUE                                                          ØUT02110
      V(K,I)=FFØR(I)                                                    ØUT02120
      W(K,I)=RREV(I)                                                    ØUT02130
   50 CØNTINUE                                                          ØUT02140
      WRITE(6,43) (K,(FFØR(I),I=ISTART,ISTØP))                          ØUT02150
   43 FØRMAT(1X,I2,1X,16E8.4)                                           ØUT02160
      WRITE(6,43) (K,(RREV(I),I=ISTART,ISTØP))                          ØUT02170
   39 CØNTINUE                                                          ØUT02180
      ICØUNT=ICØUNT+1                                                   ØUT02190
      IF(ICØUNT.EQ.2) GØ TØ 45                                          ØUT02200
      ISTART=20                                                         ØUT02210
      GØ TØ 46                                                          ØUT02220
   45 CØNTINUE                                                          ØUT02230
      DØ 85 K=1,KRATE                                                   ØUT02240
      XMAX(K)=.0                                                        ØUT02250
      YMAX(K)=.0                                                        ØUT02260
      DØ 86 I=2,NP2                                                     ØUT02270
      IF(V(K,I).GT.XMAX(K)) GØ TØ 72                                    ØUT02280
      GØ TØ 73                                                          ØUT02290
   72 XMAX(K)=V(K,I)                                                    ØUT02300
      IX(K)=I                                                           ØUT02310
   73 IF(W(K,I).GT.YMAX(K)) GØ TØ 74                                    ØUT02320
      GØ TØ 86                                                          ØUT02330
   74 YMAX(K)=W(K,I)                                                    ØUT02340
      IY(K)=I                                                           ØUT02350
   86 CØNTINUE                                                          ØUT02360
   85 CØNTINUE                                                          ØUT02370
      IF(IFINK.EQ.0) GØ TØ 204                                          ØUT02380
      WRITE(6,205)                                                      ØUT02390
  205 FØRMAT(1H1,'MAXIMUM HEAT RELEASE RATES FØR EACH REACTIØN')        ØUT02400
      GØ TØ 206                                                         ØUT02410
  204 CØNTINUE                                                          ØUT02420
      WRITE(6,75)                                                       ØUT02430
   75 FØRMAT(1H1,'MAXIMUM RATES ØF FØRWARD AND REVERSE REACTIØNS')      ØUT02440
```

```
206 CØNTINUE                                                              ØUT02450
    WRITE(6,69)                                                           ØUT02460
    WRITE(6,76)                                                           ØUT02470
 76 FØRMAT(1X,'PØSITIØN,FØRWARD MAX  PØSITIØN,REVERSE MAX  REACTIØN')     ØUT02480
    WRITE(6,77) (IX(K),XMAX(K),IY(K),YMAX(K),K,K=1,KRATE)                 ØUT02490
 77 FØRMAT(4X,I2,5X,E8.4,7X,I2,5X,E8.4,7X,I2)                             ØUT02500
    WRITE(6,84)                                                           ØUT02510
 84 FØRMAT(1H1,'RANK',2X,'REACTIØN',2X,'DIRECTIØN',2X,'MAXIMUM')          ØUT02520
    DØ 78 K=1,KRATE                                                       ØUT02530
    Z(K)=XMAX(K)                                                          ØUT02540
 78 Z(K+KRATE)=YMAX(K)                                                    ØUT02550
    KK=2*KRATE                                                            ØUT02560
    ICØUNT=0                                                              ØUT02570
 83 ZTAB=.0                                                               ØUT02580
    DØ 79 K=1,KK                                                          ØUT02590
    IF(SKIP(K)) GØ TØ 79                                                  ØUT02600
    IF(Z(K).GT.ZTAB) GØ TØ 80                                             ØUT02610
    GØ TØ 79                                                              ØUT02620
 80 ZTAB=Z(K)                                                             ØUT02630
    KTAB=K                                                                ØUT02640
 79 CØNTINUE                                                              ØUT02650
    IF(KTAB.LE.KRATE) GØ TØ 87                                            ØUT02660
    TAB=3HREV                                                             ØUT02670
    NTAB=KTAB-KRATE                                                       ØUT02680
    GØ TØ 88                                                              ØUT02690
 87 TAB=3HFØR                                                             ØUT02700
    NTAB=KTAB                                                             ØUT02710
 88 CØNTINUE                                                              ØUT02720
    ICØUNT=ICØUNT+1                                                       ØUT02730
    WRITE(6,81) ICØUNT,NTAB,TAB,ZTAB                                      ØUT02740
 81 FØRMAT(2X,I2,6X,I2,7X,A3,6X,E8.4)                                     ØUT02750
    SKIP(KTAB)=.TRUE.                                                     ØUT02760
    IF(ICØUNT.EQ.KK) GØ TØ 82                                             ØUT02770
    GØ TØ 83                                                              ØUT02780
 82 CØNTINUE                                                              ØUT02790
```

```
      IF(.NØT.FLAG(11)) GØ TØ 99                                        ØUT02800
      DØ 200 K=1,KRATE                                                  ØUT02810
      DØ 200 I=2,NP2                                                    ØUT02820
      V(K,I)=HV(K,I)                                                    ØUT02830
  200 W(K,I)=HW(K,I)                                                    ØUT02840
      ICØUNT=0                                                          ØUT02850
      ISTART=4                                                          ØUT02860
  209 ISTØP=ISTART+15                                                   ØUT02870
      WRITE(6,201)                                                      ØUT02880
  201 FØRMAT(1H1,'HEAT RELEASE RATES FØR EACH REACTIØN')                ØUT02890
      WRITE(6,69)                                                       ØUT02900
      WRITE(6,44) (I,I=ISTART,ISTØP)                                    ØUT02910
      DØ 202 K=1,KRATE                                                  ØUT02920
      WRITE(6,250) (K,(HV(K,I),I=ISTART,ISTØP))                         ØUT02930
      WRITE(6,250) (K,(HW(K,I),I=ISTART,ISTØP))                         ØUT02940
  250 FØRMAT(1X,I2,1X,16E8.3)                                           ØUT02950
  202 CØNTINUE                                                          ØUT02960
      ICØUNT=ICØUNT+1                                                   ØUT02970
      IF(ICØUNT.EQ.2) GØ TØ 99                                          ØUT02980
      ISTART=20                                                         ØUT02990
      GØ TØ 209                                                         ØUT03000
   99 RETURN                                                            ØUT03010
      END                                                               ØUT03020
```

REFERENCES

1) Patankar, S. V. and Spalding, D. B., Heat and Mass Transfer in Boundary Layers (Intertext Books, London, 1970).

2) Lovachev, L. A. and Kaganova, Z. I., Dokl. Akad. Nauk SSSR 188, 1087 (1969).

3) Spalding, D. B. and Stephenson, P. L., Proc. Roy. Soc. Lond. A324, 315 (1971).

4) Hirschfelder, J. O., Curtiss, C. F., and Bird, R. B., Molecular Theory of Gases and Liquids (Wiley, New York, 1954).

5) Lindsay, A. L. and Bromley, L. A., Ind. Eng. Chem. 42, 1508 (1950).

APPENDIX A.

DERIVATION OF THE ONE-DIMENSIONAL LAMINAR FLAME EQUATIONS

Consider a control volume δV oriented as shown in Fig. A1. Since the flame is one-dimensional, there will be gradients only in the y-direction. The gas mixture flows from right to left through the + face of the control volume and out through the - face. The area of each face is δA, and the length of the volume is δy. The velocity of the mixture v is the mass average flow velocity defined by the equation

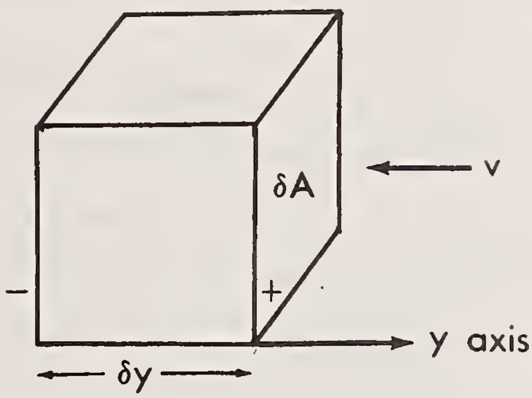


Figure A1.
Orientation of control volume

$$v = \rho^{-1} \sum_j C_j M_j \bar{v}_j$$

where $\bar{v}_j$ is the average velocity, C_j the concentration, and M_j the molecular weight of each chemical species; ρ is the density of the mixture.

There are three conservation equations to be derived;

a) the conservation of total mass;

b) the conservation of each chemical species;

c) the conservation of energy.

a) <u>Conservation of total mass</u>

Let ρ_+ be the density of the gas mixture entering the + face of δV, and let ρ_- be the density of the gas leaving at the - face. Let v_+ and v_- be the mass average velocities of the gas at the + and - faces. If ρ is in units of kg/m^3 and v in m/s, then the number of kg of gas entering δV per second is $\delta A v_+ \rho_+$, and the amount leaving is

$\delta A v_- \rho_-$. The difference between these two quantities gives the rate at which the mass of gas in δV changes. This is $\delta\rho\delta V/\delta t$, where $\delta\rho$ is the change in the average density of the gas in δV in the time interval δt. Thus we get the equation,

$$\delta\rho\delta V/\delta t = - \delta A(v_+\rho_+ - v_-\rho_-).$$

Note that v_+ and v_- are negative numbers because the flow is from right to left. Dividing this equation by δV, we obtain the expression

$\delta\rho/\delta t = - (v_+\rho_+ - v_-\rho_-)/\delta y$, since $\delta V = \delta A\ \delta y$. If δy is sufficiently small, this equation can be written in derivative notation;

$$\partial\rho/\partial t = - \partial(\rho v)/\partial y. \qquad \text{(A1)}$$

This is the mass conservation equation.

b) Species conservation equations

Let Y_j be the mass fraction of the chemical species j. Then $Y_j\rho$ is the density of species j in the mixture. We can derive the same type of equation for the rate at which the mass of j changes in δV as we did for the change of total mass. We get,

$$\delta(\rho Y_j)\delta V/\delta t = - \delta A \left[(v + V_j)_+(\rho Y_j)_+ - (v + V_j)_-(\rho Y_j)_-\right] + R_j\delta V.$$

Here, V_j is the diffusion velocity of species j, and R_j is the mass production rate of species j arising from chemical reactions in δV. Dividing this equation by δV and passing over to derivative notation, we get the species conservation equations,

$$\partial(\rho Y_j)/\partial t = -\frac{\partial}{\partial y}\left[(v + V_j)(\rho Y_j)\right] + R_j, \quad j=1,n-1 \cdot \qquad \text{(A2)}$$

There will be one of these equations for each species except one. The mass fraction of the n'th species is related to the mass fractions of the remaining species by the requirement that the sum of all the mass fractions equal unity; i.e. $\sum_j^n Y_j = 1$.

c) Energy conservation equation

Let h_j be the enthalpy per unit mass of the pure species j at the temperature T. The enthalpy of the mixture is then $h = \sum_j Y_j h_j$. Let λ be the thermal conductivity of the mixture (J $m^{-1} s^{-1} {}^{\circ}K^{-1}$). The rate at which energy enters the + face of the control volume due to heat conduction arising from a temperature gradient $(\partial T/\partial y)_+$ at the + face is $(\lambda \partial T/\partial y)_+ \delta A$. In addition, energy is carried into δV through the + face by species j. This occurs at a rate $(v + V_j)_+ (\rho Y_j h_j)_+ \delta A$. Equating the rate of change in the total energy in δV to the difference between the amount entering and leaving by heat conduction and the amounts carried in and out by each species, we get,

$$\delta(\rho h)\delta V/\delta t = \delta A \left[(\lambda \partial T/\partial y)_+ - (\lambda \partial T/\partial y)_-\right] - \delta A \left\{ \sum_j (v + V_j)_+ (\rho Y_j h_j)_+ - \sum_j (v + V_j)_- (\rho Y_j h_j)_- \right\}.$$

Dividing this equation by δV and going to derivative notation, we obtain the energy conservation equation,

$$\partial(\rho h)/\partial t = \frac{\partial}{\partial y}(\lambda \partial T/\partial y) - \frac{\partial}{\partial y}\left(\sum_j (v + V_j) Y_j \rho h_j \right). \qquad \text{(A3)}$$

APPENDIX B.

DETERMINATION OF THE VELOCITY OF A STEADILY PROPAGATING FLAME

The equations derived in Appendix A give the time dependence per unit volume of the total mass, the masses of each chemical species, and the total energy in the system. For a steadily propagating flame, the time derivatives will be zero in a coordinate system which moves with the velocity of the flame front. Let us consider the consequences of allowing the time derivatives in Eqs. A1, A2, and A3 to vanish. From A1 we get $\partial(\rho v)/\partial y = 0$; thus $\rho v = M$ is constant throughout the flame. M is the mass flow rate in the steady flame ($kg\ m^{-2}s^{-1}$).
From A2, we get,

$$\frac{\partial}{\partial y}(v + V_j)(\rho Y_j) = R_j.$$

Integrating this equation from the hot to the cold side of the flame gives,

$$\int_H^C \frac{\partial}{\partial y}(v + V_j)(\rho Y_j)\, dy = \int_H^C R_j dy = (v + V_j)_C(\rho Y_j)_C - (v + V_j)_H(\rho Y_j)_H.$$

If the integration extends sufficiently far into the hot and cold regions, the diffusion velocities $(V_j)_C$ and $(V_j)_H$ become negligible compared to v_C and v_H since the concentration gradients become small. Thus, we can write this equation as,

$$\int_H^C R_j dy = (\rho v Y_j)_C - (\rho v Y_j)_H = M(Y_{jC} - Y_{jH}),$$

since $(\rho v)_C = (\rho v)_H = M$ from the steady-state mass conservation equation A1. Any of the integrals $\int R_j dy$ can be used to calculate M or v_C, which is the steady flame velocity referred to the cold side of the flame. Thus, the equation for v_C is,

$$v_C = \int_H^C R_j dy \Big/ \left[\rho_C (Y_{jC} - Y_{jH})\right]. \qquad (A4)$$

The solution of the flame equations in the steady-state limit yields the mass fractions Y_j of each chemical species as a function of y, the distance through the flame. From these values the mass production rates

R_j can be calculated as a function of y. If the flame equations have been solved correctly, each of the integrals $\int R_j dy$ should yield the same value for v_C.

Consider now an adiabatic steadily propagating flame. In this situation there is no heat conducted through the cold or hot boundaries. Thus the temperature gradients vanish at the boundaries; i.e., $(\partial T/\partial y)_C = (\partial T/\partial y)_H = 0$. From this, together with the condition $(V_j)_C = (V_j)_H = 0$, we get from the steady-state form of the energy equation A3, the relation,

$$\sum_j (\rho v Y_j h_j)_C = \sum_j (\rho v Y_j h_j)_H.$$

Since ρv is a constant this becomes,

$$\sum_j (Y_j h_j)_C = \sum_j (Y_j h_j)_H, \quad \text{or simply,} \quad h_C = h_H.$$

Therefore, in an adiabatic flame, the enthalpy of the gas mixture is the same at the hot and cold boundaries. Inside the flame, however, the enthalpy will generally be different from h_C because $\partial T/\partial y$ and the diffusion velocities V_j are not zero there. There exists, however, a condition in which h remains constant throughout the flame. If the energy flux from heat conduction $\lambda\, \partial T/\partial y$ equals the energy flux from diffusion $\rho \sum_j V_j Y_j h_j$ at each point in the flame, then we see from A3 that the quantity $\rho v h$ is constant throughout the flame. Since we know from A1 that ρv is invarient, then h must also be invarient. This condition is equivalent to assuming that the Lewis number $D\rho C_p/\lambda$ for the mixture is equal to unity at all points in the flame. The reason for this equivalence will be discussed in Appendix G. Although this condition is only approximately satisfied by real flames, it is often envoked to avoid having to solve the energy equation.

APPENDIX C.

CALCULATION OF DIFFUSION VELOCITIES

To solve the species conservation equations it is necessary to assume that Fick's law adequately represents the mass diffusion processes in the flame. This law assumes that a particular diffusion velocity V_j is proportional to the concentration gradient of species j only. If we consider each species in the mixture in turn to be one component of a binary mixture, with all the other species lumped together as the other component, then it is possible to derive an expression for the proportionality factor in Fick's law from the kinetic theory of gases. This factor will be a function of all the possible binary diffusion coefficients in the mixture and the concentrations of all the species.

According to Fick's law, the diffusion velocity of species j is given by the equation,

$$V_j = -\Delta_j x_j^{-1} \partial x_j/\partial y, \tag{A5}$$

where Δ_j is an effective diffusion coefficient for species j in the mixture, and x_j is its mole fraction. Our task is to find an expression for Δ_j in terms of the mole fractions x_j of all the species and the binary diffusion coefficients D_{ji} for all possible pairs of species in the mixture.

The kinetic theory of gases[4] gives the following expression for $\partial x_j/\partial y$ in terms of the binary diffusion coefficients and the diffusion velocities of the different species j;

$$\frac{\partial x_j}{\partial y} = x_j \sum_{i \neq j} \frac{x_i}{D_{ji}} (V_i - V_j) = x_j \left[\sum_{i \neq j} \frac{x_i V_i}{D_{ji}} - V_j \sum_{i \neq j} \frac{x_i}{D_{ji}} \right]. \quad j=1,n \tag{A6}$$

These equations are subject to the condition,

$$\sum_j x_j V_j = 0. \tag{A7}$$

We now assume that all of the diffusion velocities except V_j have the same value V^*. Then A6 becomes,

$$\partial x_j/\partial y = x_j(V^* - V_j)\sum_{i\neq j} x_i/D_{ji}. \tag{A8}$$

From A7 we get the relation,

$$V^* = -x_j V_j (1 - x_j)^{-1}.$$

Substituting this expression for V^* into A8 gives,

$$\frac{\partial x_j}{\partial y} = -\left(\frac{x_j}{1 - x_j}\right) V_j \sum_{i\neq j} \frac{x_i}{D_{ji}} .$$

Solving for V_j, we get

$$V_j = -\frac{(1 - x_j)}{\sum_{i\neq j} \frac{x_i}{D_{ji}}} \frac{1}{x_j} \frac{\partial x_j}{\partial y} .$$

The proportionality factor Δ_j in Fick's law is thus,

$$\Delta_j = \frac{(1 - x_j)}{\sum_{i\neq j} \frac{x_i}{D_{ji}}} . \tag{A9}$$

APPENDIX D.

CALCULATION OF THE THERMAL CONDUCTIVITY OF THE FLAME

An empirical equation was used to calculate the thermal conductivity of the mixture of gases in the flame. This was developed by Lindsay and Bromley[5] It requires a knowledge of the pure component conductivities, heat capacities, boiling points, and molecular weights. The expression for λ is,

$$\lambda = \sum_i \lambda_i \left[x_i \sum_i A_{ij} x_j \right]^{-1}, \tag{A10}$$

where λ_i is the thermal conductivity of species i, and the quantities A_{ij} are given by the formula,

$$A_{ij} = \frac{1}{4} \left\{ 1 + \left[\frac{\mu_i}{\mu_j} \left(\frac{M_j}{M_i} \right)^{\frac{3}{4}} \frac{\left(1 + \frac{S_i}{T}\right)}{\left(1 + \frac{S_j}{T}\right)} \right]^{\frac{1}{2}} \right\}^2 \frac{\left(1 + \frac{S_{ij}}{T}\right)}{\left(1 + \frac{S_i}{T}\right)}. \tag{A11}$$

The viscosity ratio μ_i/μ_j is given by the expression,

$$\mu_i/\mu_j = (\lambda_i/\lambda_j)(C_{pj} + 1.25R/M_j)(C_{pi} + 1.25R/M_i)^{-1},$$

where C_{pi} and C_{pj} are the heat capacities, and M_i and M_j the molecular weights of species i and j. R is the gas constant. The quantities S_i and S_j are the Sutherland constants of i and j. For hydrogen, the value is 79° K. For other pure gases it is taken to be $1.5T_B$ where T_B is the boiling point at 1 atmosphere pressure. The constant S_{ij} is the geometric mean of S_i and S_j except when one gas is strongly polar; then $S_{ij} = 0.733(S_i S_j)^{-\frac{1}{2}}$.

APPENDIX E.

VON MISES TRANSFORMATION OF THE FLAME EQUATIONS

Preliminaries

Before applying the von Mises transformation to the flame equations let us insert Fick's diffusion formula into the species conservation equation and eliminate temperature from the energy equation.

Substitution of A5, the expression for V_j, into the species conservation equations A2 yields the equation,

$$\rho \frac{\partial Y_j}{\partial t} + \rho v \frac{\partial Y_j}{\partial y} = \frac{\partial}{\partial y}\left(\Delta_j \rho \frac{Y_j}{x_j} \frac{\partial x_j}{\partial y}\right) + R_j .$$

Using the identity,

$$\frac{Y_j}{x_j} \frac{\partial x_j}{\partial y} = \frac{\partial Y_j}{\partial y} + \frac{Y_j}{\langle M \rangle} \frac{\partial \langle M \rangle}{\partial y} ,$$

where $\langle M \rangle = \sum_j x_j M_j$ is the average molecular weight of the mixture we obtain,

$$\rho \frac{\partial Y_j}{\partial t} + \rho v \frac{\partial Y_j}{\partial y} = \frac{\partial}{\partial y}\left(\Delta_j \rho \frac{\partial Y_j}{\partial y}\right) + \frac{\partial}{\partial y}\left(\Delta_j \rho \frac{Y_j}{\langle M \rangle} \frac{\partial \langle M \rangle}{\partial y}\right) + R_j . \qquad \text{(A12)}$$

Consider next the energy equation (A3). The enthalpy h is a function of the mass fractions Y_j and the temperature T; i.e., $h = f(Y_j, T)$. The partial derivative of h with respect to y is,

$$\frac{\partial h}{\partial y} = \sum_j \frac{\partial h}{\partial Y_j} \frac{\partial Y_j}{\partial y} + \frac{\partial h}{\partial T} \frac{\partial T}{\partial y} = \sum_j \frac{\partial Y_j}{\partial y} h_j + C_p \frac{\partial T}{\partial y} , \qquad \text{(A13)}$$

where we used the relations $\partial h / \partial Y_j = h_j$ and $\partial h / \partial T = C_p$, the heat capacity of the mixture at constant pressure. Solving A13 for $\partial T / \partial y$, and substituting this as well as A5 into the energy conservation eqation A3, we get after some algebraic manipulation,

$$\rho \frac{\partial h}{\partial t} + \rho v \frac{\partial h}{\partial y} = \frac{\partial}{\partial y}\left(\frac{\lambda}{C_p} \frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial y}\left\{\sum h_j \left[\frac{\Delta_j \rho}{\langle M \rangle} \frac{\partial (Y_j \langle M \rangle)}{\partial y} - \frac{\lambda}{C_p} \frac{\partial Y_j}{\partial y}\right]\right\} . \qquad \text{(A14)}$$

We now change from mass fractions to a different concentration variable φ_j defined as

$$\varphi_j \equiv Y_j/M_j. \tag{A15}$$

This has units of moles/kg. The species and energy conservation equations then take the form,

$$\rho\frac{\partial\varphi_j}{\partial t} + \rho v\frac{\partial\varphi_j}{\partial y} = \frac{\partial}{\partial y}\left(\Delta_j\rho\frac{\partial\varphi_j}{\partial y}\right) + \frac{\partial}{\partial y}\left(\Delta_j\rho\,\frac{\varphi_j}{\langle M\rangle}\,\frac{\partial\langle M\rangle}{\partial y}\right) + \frac{R_j}{M_j} \tag{A16}$$

$$\rho\frac{\partial h}{\partial t} + \rho v\frac{\partial h}{\partial y} = \frac{\partial}{\partial y}\left(\frac{\lambda}{C_p}\frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial y}\left\{\sum_j h_j^*\left[\frac{\Delta_j\rho}{\langle M\rangle}\,\frac{\partial(\varphi_j\langle M\rangle)}{\partial y} - \frac{\lambda}{C_p}\frac{\partial\varphi_j}{\partial y}\right]\right\} \tag{A17}$$

In A17 the quantities $h_j^* = h_jM_j$ are the enthalpies of the pure species in J/mole.

Von Mises Transformation

We introduce a new spatial variable ψ in place of y. It is defined by the equations,

$$\partial\psi/\partial y = \rho, \quad \text{and} \quad \partial\psi/\partial t = -\rho v. \tag{A18}$$

Applying this transformation to A16 and A17 eliminates the convection terms $\rho v\partial\varphi_j/\partial y$ and $\rho v\partial h/\partial y$. The mass conservation equation A1 is satisfied automatically because of the relations,

$$\partial\rho/\partial t = \partial^2\psi/\partial t\partial y = \partial^2\psi/\partial y\partial t = -\partial(\rho v)/\partial y.$$

We are left with only the species and energy equations in the form,

$$\frac{\partial\varphi_j}{\partial t} = \frac{\partial}{\partial\psi}\left(\Delta_j\rho^2\,\frac{\partial\varphi_j}{\partial\psi}\right) + \frac{\partial}{\partial\psi}\left(\Delta_j\rho^2\frac{\varphi_j}{\langle M\rangle}\,\frac{\partial\langle M\rangle}{\partial\psi}\right) + \frac{R_j}{\rho M_j} \tag{A19}$$

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial\psi}\left(\frac{\lambda\rho}{C_p}\,\frac{\partial h}{\partial\psi}\right) + \frac{\partial}{\partial\psi}\left\{\sum_j \frac{h_j^*\Delta_j\rho^2}{\langle M\rangle}\,\frac{\partial(\varphi_j\langle M\rangle)}{\partial\psi} - \frac{\lambda\rho}{C_p}\sum_j h_j^*\,\frac{\partial\varphi_j}{\partial\psi}\right\} \tag{A20}$$

APPENDIX F.

SPALDING'S TRANSFORMATION OF THE FLAME EQUATIONS

The technique developed by Patankar and Spalding for solving the two-dimensional boundary-layer equations can be applied to the solution of the flame equations. To apply this method, a new spatial variable ω is introduced. This is defined by the equation,

$$\omega = (\psi - \psi_H)/(\psi_C - \psi_H). \tag{A21}$$

The quantities ψ_C & ψ_H are the values of ψ at the cold and hot boundaries of the flame. They are functions of time only. The advantage of using ω is that if the calculation procedure is limited to the range $0 \leqq \omega \leqq 1$, then it is automatically limited to that part of space where the important changes in the dependent variables φ_j and h occur.

After the transformation, the species and energy equations A19 and A20 become,

$$\frac{\partial\varphi_j}{\partial t} + \left(\frac{a + b\omega}{\eta}\right)\frac{\partial\varphi_j}{\partial\omega} = \frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left(\Delta_j\rho^2\frac{\partial\varphi_j}{\partial\omega}\right) + \frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left(\Delta_j\rho^2\frac{\varphi_j}{\langle M\rangle}\frac{\partial\langle M\rangle}{\partial\omega}\right) + \frac{R_j}{\rho M_j} \tag{A22}$$

$$\frac{\partial h}{\partial t} + \left(\frac{a + b\omega}{\eta}\right)\frac{\partial h}{\partial\omega} = \frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left(\frac{\lambda\rho}{C_p}\frac{\partial h}{\partial\omega}\right)$$

$$+ \frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left\{\sum_j h_j^*\left[\frac{\Delta_j\rho^2}{\langle M\rangle}\frac{\partial(\varphi_j\langle M\rangle)}{\partial\omega} - \frac{\lambda\rho}{C_p}\frac{\partial\varphi_j}{\partial\omega}\right]\right\} \tag{A23}$$

where $\eta = \psi_C - \psi_H$. In making the transformation we have used the relation for $\partial\omega/\partial t$ in the ψ,t coordinate system, $\partial\omega/\partial t = (a + b\omega)/\eta$. The quantities a and b are defined by,

$$a = - d\psi_H/dt, \qquad b = - d\psi_C/dt + d\psi_H/dt.$$

The derivatives $d\psi_C/dt$ and $d\psi_H/dt$ are the mass flow rates across the cold and hot boundaries. In the steady-state flame they have the same constant

value. When this limit is reached $\partial\varphi_j/\partial t \to 0$, $b \to 0$, and A22, integrated from $\omega = 0$ to 1, becomes,

$$\rho_C v_C(\varphi_C - \varphi_H) = \eta\int_0^1 \frac{R_j}{\rho M_j}\, d\omega \quad . \tag{A24}$$

This relation arises because $a = \rho_H v_H = \rho_C v_C$ in the steady-state limit (see A18); also the gradients of φ_j and $\langle M \rangle$ are assumed to vanish at the hot and cold boundaries. This gives an equation for v_C in terms of R_j expressed in the ω, t coordinate system which is analogous to A4 derived for R_j in the y, t system. For v_C we have,

$$v_C = \left(\eta \int_0^1 \frac{R_j}{\rho} d\omega\right)\left[\rho_C M_j(\varphi_C - \varphi_H)\right]^{-1}. \tag{A25}$$

APPENDIX G.

UNITY LEWIS NUMBER APPROXIMATION

In Appendix B we demonstrated that in the adiabatic flame, if the energy flux from heat conduction equals that from species diffusion at each point in the flame, then the enthalpy remains constant throughout the flame. We stated that this assumption was equivalent to the assumption that the Lewis number was unity at all points. We shall now demonstrate how this equivalence arises.

The Lewis number is defined for all species pairs as,

$$Le_{ij} \equiv \rho C_p D_{ij}/\lambda,$$

where D_{ij} is the binary diffusion coefficient for species pair i,j and C_p is the average specific heat at constant pressure for the mixture. If we assume that Le = 1 for all species pairs, then we must assume that all the binary diffusion coefficients D_{ij} are equal. Suppose that $D_{ij} = D$. When this is substituted into A9, the expression for Δ_j, we get,

$$\Delta_j = \frac{1 - x_j}{\frac{1}{D}\sum_{i \neq j} x_i} = D, \qquad \text{since } 1 - x_j = \sum_{i \neq j} x_i.$$

Thus, the assumption Le = 1 gives,

$$\rho C_p D/\lambda = \rho C_p \Delta_j/\lambda = 1; \quad \text{or} \quad D\rho^2 = \Delta_j \rho^2 = \lambda\rho/C_p.$$

Consider the energy equation A23 in the ω,t coordinate system under steady-state conditions. Here, $\partial h/\partial t = 0$, b = 0, and h is a function of ω only. We must make one further assumption, namely that the change in the average molecular weight $\langle M \rangle$ is negligible throughout the flame. Then the last term in A23 vanishes when $\Delta_j \rho^2 = \lambda\rho/C_p$ and in the steady-state A23 becomes,

$$a \frac{dh}{d\omega} = \frac{1}{\eta} \frac{d}{d\omega}\left(\frac{\lambda\rho}{C_p} \frac{dh}{d\omega}\right). \tag{A26}$$

Next, insert into A26 the variable $U \equiv dh/d\omega$. We get,

$$aU = \frac{1}{\eta}\frac{d}{d\omega}\left(\frac{\lambda\rho}{C_p}\right)U + \frac{1}{\eta}\frac{\lambda\rho}{C_p}\frac{dU}{d\omega}\,.$$

Solving for $dU/d\omega$ we obtain,

$$\frac{1}{U}\frac{dU}{d\omega} = \frac{\eta C_p}{\lambda\rho}\left[a - \frac{1}{\eta}\frac{d}{d\omega}\left(\frac{\lambda\rho}{C_p}\right)\right] \equiv f(\omega).$$

Integrating this expression starting from the cold side of the flame we get,

$$\ln(U/U_0) = \int_0^{\omega} f(\omega)d\omega\ , \quad \text{or } U = U_0 \exp\left[\int_0^{\omega} f(\omega)d\omega\right], \tag{A27}$$

where U_0 is the value of $dh/d\omega$ at the cold boundary. For the adiabatic flame $U_0 = 0$ at the cold side; from A27 we see that if $U_0 = 0$, then $U = dh/d\omega$ is zero for all values of ω. Thus h remains unchanged throughout the flame.

Therefore, if the enthalpy is to remain constant in the adiabatic flame we must assume that $\Delta_j\rho^2 = \lambda\rho/C_p$ and that $\frac{d\langle M\rangle}{d\omega} = 0$; it is not necessary to assume that $\lambda\rho/C_p$ remains constant.

APPENDIX H.

FINITE-DIFFERENCE FORMS OF THE FLAME EQUATIONS

The species and energy conservation equations A22 and A23 have the same general form;

$$\frac{\partial \Phi}{\partial t} + (\alpha + \beta\omega)\frac{\partial \Phi}{\partial \omega} = \frac{\partial}{\partial \omega}\left(\gamma \frac{\partial \Phi}{\partial \omega}\right) + \mathcal{S}_m + \mathcal{S}, \tag{A28}$$

where $\alpha = a/\eta$, $\beta = b/\eta$, γ is the transport coefficient, $\mathcal{S}_m$ is a source term arising from the diffusion of average molecular weight, and $\mathcal{S}$ is a source term arising from chemical reaction in the species equation, and from the difference between conductive and diffusional transport of enthalpy in the energy equation. γ, $\mathcal{S}_m$, and $\mathcal{S}$ are shown explicitly below;

	Energy	Species
γ	$\frac{1}{\eta^2}\frac{\lambda\rho}{C_p}$	$\frac{1}{\eta^2}\Delta_j\rho^2$
$\mathcal{S}_m$	$\frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left(\sum_j h_j^* \Delta_j\rho^2 \frac{\varphi_j}{\langle M\rangle}\frac{\partial\langle M\rangle}{\partial\omega}\right)$	$\frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left(\Delta_j\rho^2\frac{\varphi_j}{\langle M\rangle}\frac{\partial\langle M\rangle}{\partial\omega}\right)$
$\mathcal{S}$	$\frac{1}{\eta^2}\frac{\partial}{\partial\omega}\left[\sum_j h_j^*\left(\Delta_j\rho^2 - \frac{\lambda\rho}{C_p}\right)\frac{\partial\varphi_j}{\partial\omega}\right]$	$\frac{R_j}{\rho M_j}$

To set up a finite-difference equation for A28, we divide the ω axis between 0 and 1 into N strips which need not have the same widths. The profile of the dependent variable Φ is assumed to be linear between grid points. We consider control volumes whose boundaries lie midway between the grid points. A portion of the grid is shown in Fig. A2. This shows the control volume and illustrates the linear-profile assumption. Equation A28 is now integrated over the control volume shown in Fig. A2. This leads to the following equation;

$$\left[\int_{-}^{+}\Phi\, d\omega - \int_{-}^{+}\Phi_P d\omega\right]/\delta t + \left[\{(\alpha + \beta\omega)\Phi\}_+ - \{(\alpha + \beta\omega)\Phi\}_-\right] - \beta\int_{-}^{+}\Phi\, d\omega$$

$$= \left[\left(\gamma\frac{\partial\Phi}{\partial\omega}\right)_+ - \left(\gamma\frac{\partial\Phi}{\partial\omega}\right)_-\right] + \int_{-}^{+}\mathcal{S}_m d\omega + \int_{-}^{+}\mathcal{S}\, d\omega \qquad (A29)$$

The subscript P on Φ is used to denote its value before the time step δt is taken. An unscripted Φ refers to its value after the time step.

It can be shown after some surprisingly lengthly algebra that the integral over the control volume of any function like Φ, which is assumed to be linear between grid points has the value,

$$\int_{-}^{+}\Phi d\omega = \tfrac{1}{8}(3\Omega\Phi + \Omega_+\Phi_{++} + \Omega_-\Phi_{--}), \qquad (A30)$$

where $\Omega = \omega_{i+1} - \omega_{i-1}$,

$\Omega_+ = \omega_{i+1} - \omega_i$

$\Omega_- = \omega_i - \omega_{i-1}$.

Note that $\omega_+ = \frac{1}{2}(\omega_{i+1} + \omega_i)$ and $\omega_- = \frac{1}{2}(\omega_i + \omega_{i-1})$. We introduce some additional symbols

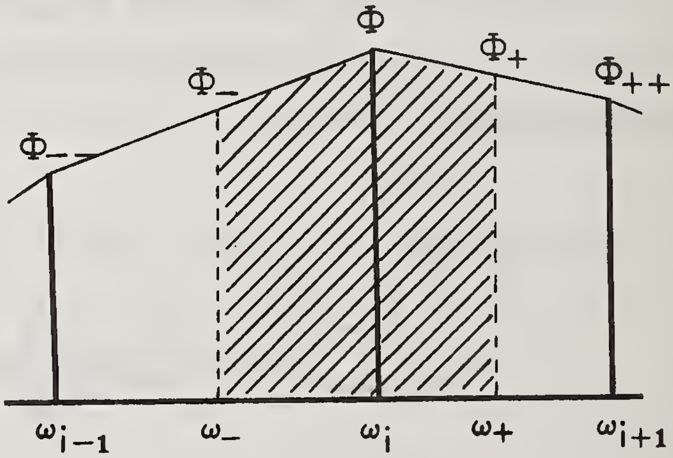


Figure A2.

$P \equiv \eta/\delta t$

$G \equiv -\eta\beta = b$

$L_+ \equiv \eta(\alpha + \beta\omega_+) = a - \omega_+ G$

$L_- \equiv a - \omega_- G \qquad (A31)$

The derivatives of Φ at the control volume boundaries are approximated by

$$\left(\frac{\partial\Phi}{\partial\omega}\right)_+ = \frac{\Phi_{++} - \Phi}{\omega_{i+1} - \omega_i}, \quad \text{and} \quad \left(\frac{\partial\Phi}{\partial\omega}\right)_- = \frac{\Phi - \Phi_{--}}{\omega_i - \omega_{i-1}} \qquad (A32)$$

Next, we define T_+ and T_- by the equations

$$\eta\left(\gamma\frac{\partial\Phi}{\partial\omega}\right)_+ = T_+(\Phi_{++} - \Phi), \quad \text{and} \quad \eta\left(\gamma\frac{\partial\Phi}{\partial\omega}\right)_- = T_-(\Phi - \Phi_{--}). \qquad (A33)$$

Thus we have

$$T_+ = \eta\gamma/(\omega_{i+1} - \omega_i) \quad \text{and} \quad T_- = \eta\gamma/(\omega_i - \omega_{i-1}). \qquad (A34)$$

We further define

$$S_m \equiv \eta \int_-^+ \mathcal{S}_m d\omega \qquad \text{and } S \equiv \eta \int_-^+ \mathcal{S} d\omega. \qquad (A35)$$

For the species chemical source term $\mathcal{S}$ only, we take S to have the form $S(\text{species}) = S_P + S_F\Phi$, where Φ is the value after the time step δt is taken. S_P and S_F will be functions of all the Φ for the different species, but the values will be taken as the prestep ones. For all of the other source terms, only the prestep values of the variables are used. Thus for these, $S_F = 0$.

Let us then generalize S_P somewhat, and consider it to be compounded as follows;

$$S_P = S_m(\text{species}) + S_m(\text{energy}) + S(\text{energy}) + S_P(\text{species})$$

$$S_F = S_F(\text{species}) \qquad (A36)$$

Thus,

$$S_m + S = S_P + S_F\Phi \;, \qquad (A37)$$

where S_P and S_F are defined by A36.

We now multiply A29 by η and insert into it the quantities defined by A30, A31, A34, A35, and A37. The resulting equation is linear in the poststep values of Φ; it can be solved for Φ at a particular grid point in terms of the values of Φ at the grid points on each side of it. The equation one obtains has the form

$$\Phi = A\,\Phi_{++} + B\,\Phi_{--} + C. \qquad (A38)$$

The coefficients A, B, and C contain only prestep values of Φ , and are given by

$$A \equiv A'/D$$

$$B \equiv B'/D$$

$$C \equiv C'/D$$

$$A' \equiv 2T_+ - L_+ - \tfrac{1}{4}(P + G)\Omega_+ \qquad \text{(A40)}$$

$$B' \equiv 2T_- + L_- - \tfrac{1}{4}(P + G)\Omega_-$$

$$C' \equiv \tfrac{1}{4}P(3\,\Phi_P\Omega + \Phi_{P++}\Omega_+ + \Phi_{P--}\Omega_-) + 2S_P$$

$$D \equiv 2(T_+ + T_-) + L_+ - L_- + \tfrac{3}{4}(P + G)\Omega - 2S_F$$

$$= A' + B' + P\Omega - 2S_F$$

According to Spalding, these finite difference coefficients have a defect. This can appear if $\frac{1}{2}|L_\pm|$, (the convective flux terms), become greater than $T_\pm$, (the diffusive flux terms). It is not clear to me at the present time why this problem exists. Spalding does discuss the problem at length, so the reader can look there for an explanation. The defect is remedied by replacing $T_\pm$ in the formulas A40 with the quantity $T^*_\pm$ defined by,

$$T^*_\pm \equiv \tfrac{1}{2}\left[T_\pm + |\tfrac{1}{2}L_\pm| + \left|T_\pm - |\tfrac{1}{2}L_\pm|\right|\right]. \qquad \text{(A41)}$$

This has the following consequences;

i) If $|\frac{1}{2}L_\pm| \leq T_\pm$, then $2T^*_\pm \mp L_\pm = 2T_\pm \mp L_\pm$.

Thus $T^*_\pm = T_\pm$ and the replacement of $T_\pm$ by $T^*_\pm$ has no effect.

ii) If $\frac{1}{2}L_\pm > T_\pm$, then $2T^*_+ - L_+ = 0$ and $2T^*_- + L_- = 2L_-$.

iii) If $\frac{1}{2}L_\pm < -T_\pm$, then $2T^*_+ - L_+ = -2L_+$ and $2T^*_- + L_- = 0$.

This modification of the formulas A40 has the effect of neglecting the diffusion term in the flame equations whenever it becomes somewhat smaller than the convection term.

Let us now see how the difference equations A38 are solved.

Consider the equation for grid point i;

$$\Phi_i = A_i \Phi_{i+1} + B_i \Phi_{i-1} + C_i . \qquad (A42)$$

We consider N+1 grid points. (Remember that N is the number of strips into which the ω axis is divided.) The hot boundary is arbitrarily set a grid point i = 2; thus the cold boundary will fall at i = N+2. The grid points i = 1 and N+3 will not be used in the present calculations.[@] Equations A42 can be written in the form

$$\Phi_i = A_i^* \Phi_{i+1} + B_i^*, \qquad i = 3, N+1 \qquad (A43)$$

where

$$A_3^* \equiv A_3$$

$$B_3^* \equiv B_3 \Phi_2 + C_3$$

$$A_i^* \equiv A_i/(1 - B_i A_{i-1}^*)$$

$$B_i^* \equiv (B_i B_{i-1}^* + C_i)/(1 - B_i A^*_{i-1}).$$

After calculating A_i^* and B_i^* for i = 3,N+1, one can easily calculate the Φ_i from A43 starting from $\Phi_{N+1} = A_{N+1}^* \Phi_{N+2} + B_{N+1}^*$ and working down to Φ_3. Note that the values of Φ_2 and Φ_{N+2} will be determined by the boundary conditions as discussed in Appendix K.

[@]Spalding uses the grid points i=1 and i=N+3 in the boundary-layer problem in a special way to cope with large gradients in Φ at the boundaries. Because such gradients do not arise in the flame equations, this particular treatment is unnecessary. Although we could have started the grid at i=1, this fact was not realized until later and so the computer program was written with the hot boundary starting at i=2.

APPENDIX I.

CALCULATION OF THE SOURCE TERMS

There are four source terms to be calculated; $\mathcal{S}_m$ for energy and species arises from the diffusion of average molecular weight; (these are normally quite small and can usually be neglected;) and $\mathcal{S}$ for energy and species, which arises from the difference between diffusion and conduction in the energy case, and from chemical reaction in the species case. By far the most important source term is that for the species. As described in Appendix H, this term is given special treatment. Because this term can undergo very large changes during a time step it is necessary to estimate at the beginning of a step its value at the end of the time step. The chemical source term for a particular species j is a function of the concentrations of the various species and the temperature; $\mathcal{S}_j = \mathcal{S}_j(\varphi_j, \varphi_k, \ldots\ldots, T)$. Thus we could approximate a change in $\mathcal{S}_j$ by a linear expression,

$$\delta\mathcal{S}_j \approx \frac{\partial \mathcal{S}_j}{\partial \varphi_j}\,\delta\varphi_j + \frac{\partial \mathcal{S}_j}{\partial \varphi_k}\,\delta\varphi_k + \ldots\ldots\ldots + \frac{\partial \mathcal{S}_j}{\partial T}\,\delta T.$$

Thus $(\mathcal{S}_j)_F \approx (\mathcal{S}_j)_P + \delta\mathcal{S}_j$.

Instead of using $(\mathcal{S}_j)_P$, the prestep value of $\mathcal{S}_j$, in the difference equation, we want to use $(\mathcal{S}_j)_F$, the poststep value. We should, in principle, use the above equation to calculate $\delta\mathcal{S}_j$. This, however, involves the poststep concentrations of the other species as well as j, and would make the difference equations unpleasantly complex. What we do then is neglect all of the terms but the first, and write,

$$(\mathcal{S}_j)_F \approx (\mathcal{S}_j)_P + \frac{\partial \mathcal{S}_j}{\partial \varphi_j}\,\delta\varphi_j.$$

Spalding has found this approximation to be satisfactory. In actuality, we need only a rough estimate, because when the steady-state condition is reached $\mathcal{S}_j$ ceases to change with time. In fact, it would probably be possible to solve the difference equations using only the prestep value for $\mathcal{S}_j$, but then much smaller time steps would be needed.

This treatment is unnecessary for the other source terms because they change little with time.

We shall first show how the chemical source term is calculated. A typical two reaction mechanism will be used to illustrate the method. Next, the source term for the energy flux will be calculated, and finally, the source terms arising from the diffusion of average molecular weight.

a) Calculation of the chemical source term

To show how the species source term is calculated, consider a reaction mechanism consisting of two reactions

$$A + B \underset{k_1'}{\overset{k_1}{\rightleftarrows}} C + D \qquad (1)$$

$$A + A + M \underset{k_2'}{\overset{k_2}{\rightleftarrows}} E + M \qquad (2) \qquad\qquad (A42)$$

Reaction (1) is a second order reaction in both the forward and reverse directions, while reaction (2) is third order in the forward direction and second order in the reverse. These represent the two types of reactions which we shall consider in the flame chemistry.

Consider the rate at which species A is formed. This is

$$[\dot{A}] = -k_1[A][B] + k_1'[C][D] - 2k_2[A]^2[M] + 2k_2'[E][M]. \qquad (A43)$$

The bracketed quantities signify concentrations in moles/m^3. We wish to express $\mathcal{S}(A)$, the source term for species A, in terms of the concentration variables φ (moles/kg). From the table on page A15, we see that

$$\mathcal{S}_A = R_A/(\rho M_A) \quad \text{moles kg}^{-1}\text{s}^{-1}.$$

(Remember that R_A is the mass production rate of A in kg $m^{-3}s^{-1}$.)

From a dimensional analysis, we have

$$\mathcal{S}_A = [\dot{A}]/\rho, \quad \text{and} \quad [A] = \rho\varphi_A. \tag{A44}$$

Thus we need only divide A43 by ρ and substitute for each of the concentrations their equivalent values in terms of φ; A43 becomes,

$$\mathcal{S}_A = [\dot{A}]/\rho = -k_1\rho\varphi_A\varphi_B + k_1'\rho\varphi_C\varphi_B - 2k_2\rho^2\varphi_A^2\varphi_M + 2k_2'\rho\varphi_E\varphi_M \tag{A45}$$

Let us now calculate S(species) = $\eta\int_-^+\mathcal{S}\,d\omega$. To do this we will assume that $\mathcal{S}$ is constant throughout the control volume. Thus we can write,

$$S(\text{species}) = \eta\mathcal{S}(\omega_+ - \omega_-) = \eta\mathcal{S}\tfrac{1}{2}(\omega_{++} - \omega_{--}) = \eta\mathcal{S}\tfrac{1}{2}\Omega \tag{A46}$$

Note that we could have assumed that $\mathcal{S}$ was linear between the grid points as we did for the concentration variables φ (see Fig. A2). This would have given (see A30),

$$S(\text{species}) = \eta\,\tfrac{1}{8}(3\Omega\mathcal{S} + \Omega_+\mathcal{S}_{++} + \Omega_-\mathcal{S}_{--}) \tag{A47}$$

Thus A47 uses the values of $\mathcal{S}$ at three grid points to approximate the integral $\int\mathcal{S}\,d\omega$, whereas A46 uses only the value at one point. We first work out the implications of the one point approximation in terms of our two reaction model A42. This is the approximation used in the computer program. Afterward, we shall show how the three point approximation could be incorporated into the calculation.

1) One point approximation to the chemical source term integral.

As discussed earlier in this section, we want to express the chemical source term as the sum of two terms,

$$S(\text{species}) = S_P + S_F\varphi = \tfrac{1}{2}\eta\,\mathcal{S}\Omega. \tag{A48}$$

here φ denotes the poststep value of the concentration of the particular species, e.g., species A in A45; and we use A46 to approximate S(species). Let us expand $\mathcal{S}$ in a Taylor series in φ about the point φ_P, the prestep value.

$$\begin{aligned}\mathcal{S}(\varphi) &\approx \mathcal{S}(\varphi_P) + \left(\frac{d\mathcal{S}}{d\varphi}\right)_P(\varphi - \varphi_P) = \left[\mathcal{S}(\varphi_P) - \left(\frac{d\mathcal{S}}{d\varphi}\right)_P\varphi_P\right] + \left(\frac{d\mathcal{S}}{d\varphi}\right)_P\varphi \\ &= \sigma_P + \sigma_F\varphi,\end{aligned} \tag{A49}$$

where

$$\sigma_P = \mathscr{S}(\varphi_P) - \left(\frac{d\mathscr{S}}{d\varphi}\right)_P \varphi_P \quad \text{and} \quad \sigma_F = \left(\frac{d\mathscr{S}}{d\varphi}\right)_P . \tag{A50}$$

All these terms with the P subscript have prestep values.

As an example, let us calculate σ_P and σ_F for the two reaction model A42. Taking the derivative of A45 with respect to φ_A we get,

$$\sigma_F(A) = \left[-k_1\rho\varphi_B - 4k_2\rho^2\varphi_M\varphi_A\right]_P, \tag{A51}$$

and

$$\sigma_P(A) = \left[k_1'\rho\varphi_C\varphi_D + 2k_2\rho^2\varphi_A^2\varphi_M + 2k_2'\rho\varphi_E\varphi_M\right]_P . \tag{A52}$$

The subscript P means that all of the quantities have prestep values. For this case, we thus have,

$$S(A) = \tfrac{1}{2}\eta\Omega\sigma_P(A) + \tfrac{1}{2}\eta\Omega\sigma_F(A)(\varphi_A)_F = S_P(A) + S_F(A)(\varphi_A)_F , \tag{A53}$$

where $(\varphi_A)_F$ is the poststep value of φ_A.

2) Three point approximation to the chemical source term integral.

If we were to approximate S(species) by A47, the finite difference coefficients of A40 for the species equations become,

$$A' \equiv 2T_+ - L_+ - \tfrac{1}{4}(P + G)\Omega_+ + \tfrac{1}{4}\eta\sigma_{F++}\Omega_+$$

$$B' \equiv 2T_- + L_- - \tfrac{1}{4}(P + G)\Omega_- + \tfrac{1}{4}\eta\sigma_{F--}\Omega_-$$

$$C' \equiv \tfrac{1}{4}P(3\varphi_P\Omega + \varphi_{P++}\Omega_+ + \varphi_{P--}\Omega_-)$$
$$+ 2S_m(\text{species}) + \tfrac{3}{4}\eta\sigma_P\Omega + \tfrac{1}{4}\eta\sigma_{P++}\Omega_+ + \tfrac{1}{4}\eta\sigma_{P--}\Omega_-$$

$$D \equiv 2(T_+ + T_-) + L_+ - L_- + \tfrac{3}{4}(P + G)\Omega - \tfrac{3}{4}\eta\sigma_F\Omega . \tag{A54}$$

These equations are not significantly more complicated than A40, and should be more accurate. They have not, however, been incorporated into the present computer program.

b) Calculation of the energy source term

The energy source term is easily calculated. From A35 and the table on page A15, we have

$$S = \eta \int_{-}^{+} \mathcal{S} d\omega = \frac{1}{\eta} \int_{-}^{+} \frac{\partial}{\partial \omega} \left[\sum_j h_j^* \, \Delta_j \rho^2 - \frac{\lambda\rho}{C_p} \frac{\partial \varphi_j}{\partial \omega} \right] d\omega$$

$$= \frac{1}{\eta} \left[\sum_j h_j^* \left(\Delta_j \rho^2 - \frac{\lambda\rho}{C_p} \right) \frac{\partial \varphi_j}{\partial \omega} \right]_{+} - \frac{1}{\eta} \left[\sum_j h_j^* \left(\Delta_j \rho^2 - \frac{\lambda\rho}{C_p} \right) \frac{\partial \varphi_j}{\partial \omega} \right]_{-} \qquad \text{(A55)}$$

As usual the $\pm$ subscripts refer to values at the control volume boundaries. For the first term we have

$$\left[h_j^* \left(\Delta_j \rho^2 - \frac{\lambda\rho}{C_p} \right) \right]_{+} = \tfrac{1}{2} \left[h_j^* \left(\Delta_j \rho^2 - \frac{\lambda\rho}{C_p} \right) \right]_{++} + \tfrac{1}{2} \left[h_j^* \left(\Delta_j \rho^2 - \frac{\lambda\rho}{C_p} \right) \right],$$

and for the derivative

$$\left(\frac{\partial \varphi_j}{\partial \omega} \right)_{+} = \frac{\varphi_{j++} - \varphi_j}{\omega_{++} - \omega}$$

The - boundary is treated the same way.

c) Calculation of source terms arising from diffusion of molecular weight

From A35 and the table on page A15, we have for the source terms arising from the diffusion of the average molecular weight

$$S_m = \eta \int_{-}^{+} \mathcal{S}_m d\omega = \frac{1}{\eta} \left[\sum_j h_j^* \Delta_j \rho^2 \frac{\varphi_j}{\langle M \rangle} \frac{\partial \langle M \rangle}{\partial \omega} \right]_{+} - \frac{1}{\eta} \left[\sum_j h_j^* \Delta_j \rho^2 \frac{\varphi_j}{\langle M \rangle} \frac{\partial \langle M \rangle}{\partial \omega} \right]_{-} \qquad \text{(A56)}$$

for the energy equation, and

$$S_m = \frac{1}{\eta} \left[\Delta_j \rho^2 \frac{\varphi_j}{\langle M \rangle} \frac{\partial \langle M \rangle}{\partial \omega} \right]_{+} - \frac{1}{\eta} \left[\Delta_j \rho^2 \frac{\varphi_j}{\langle M \rangle} \frac{\partial \langle M \rangle}{\partial \omega} \right]_{-} \qquad \text{(A57)}$$

for the species equation.

It is worth noting at this point that we have been somewhat careless about how we calculate quantities at the control volume boundaries. Throughout Appendix H and I, we have frequently been confronted with expressions involving products of various quantities

which are to be evaluated at the control volume boundaries. For example, let $Q_{\pm} = (q_1 q_2 q_3 \cdots\cdots q_n)_{\pm}$ where the q_k are various quantities like ρ, h_j^*, Δ_j, λ, etc. All of these quantities can be expressed in terms of the values of φ_j and h at the grid points; indeed, this is how they are evaluated at the end of every time step. This, of course, yields their values at the grid points. The values at the control volume boundaries are always taken to be the average of the values at the two grid points on each side of the boundary in question. Thus, for the + boundary we take

$$q_{k+} = \tfrac{1}{2}(q_{k++} + q_k)$$

To calculate Q_+ we often take products of the q_{k+}.

$$Q_+ = q_{1+} q_{2+} q_{3+} \cdots\cdots q_{n+}$$

Actually we should evaluate Q_+ from the expression

$$Q_+ = \tfrac{1}{2}\left[(q_1 q_2 q_3 \cdots\cdots q_n)_{++} + (q_1 q_2 q_3 \cdots\cdots q_n)\right]$$

If the q_k do not change much from grid point to grid point, the two ways of calculating Q_+ would give about the same answer.

Let us consider a case where Q is a product of two terms and compare the two methods of evaluating Q_+.

Suppose that

$$q_{1++} = q_1 + \Delta q_1 \qquad \text{and} \qquad q_{2++} = q_2 + \Delta q_2$$

The correct way to calculate Q_+ gives

$$Q_+ = \tfrac{1}{2}(q_1 q_2)_{++} + \tfrac{1}{2}(q_1 q_2) = q_1 q_2 + \tfrac{1}{2}(q_1 \Delta q_2 + q_2 \Delta q_1) + \tfrac{1}{2}\Delta q_1 \Delta q_2$$

The approximate method gives

$$Q_+(\text{appr}) = \tfrac{1}{2}(q_{1++} + q_1)\tfrac{1}{2}(q_{2++} + q_2) = q_1 q_2 + \tfrac{1}{2}(q_1 \Delta q_2 + q_2 \Delta q_1) + \tfrac{1}{4}\Delta q_1 \Delta q_2$$

Thus $Q_+(\text{appr}) = Q_+ - \tfrac{1}{4}\Delta q_1 \Delta q_2$

For Δq_k values encountered in the present calculations, the difference between the two methods of calculating products at the boundaries is negligible.

APPENDIX J

CALCULATION OF THE ENTRAINMENT RATES

The so-called entrainment rates are the mass flow rates across the hot and cold boundaries. In the steady-state flame, these will naturally be equal to each other. During the calculation, however, they are automatically adjusted to keep the grid centered on the flame front. As discussed in Appendix F, the mass flow rate across the hot boundary is

$$a = -d\psi_H/dt \equiv \dot{m}_H$$

Across the cold boundary it is

$$a + b = -d\psi_C/dt \equiv \dot{m}_C$$

Consider a position somewhere near the hot boundary; let φ_{HH} be the value of a concentration of one of the species at this point. Let φ_{CC} be its value at a point near the cold boundary. The entrainment rates are calculated from the following formulas

$$\dot{m}_C = \dot{m}_m \left[\frac{\varphi_{CC} - \varphi_C}{\varphi_H - \varphi_C} \times 100 \right]^{\nu} \tag{A58}$$

$$\dot{m}_H = \dot{m}_m \left[2 - \left\{ \frac{\varphi_H - \varphi_{HH}}{\varphi_H - \varphi_C} \times 100 \right\}^{\nu} \right] \tag{A59}$$

$$\dot{m}_m \equiv - \left| \eta \int_0^1 \frac{R_j}{\rho M_j} d\omega \right| \Big/ \left| \varphi_H - \varphi_C \right| \tag{A60}$$

where φ_C and φ_H are the values of φ at the cold and hot boundaries. Note that φ_C, φ_{CC}, φ_H, and φ_{HH} have an implied subscript j which denotes the species used to calculate the entrainment rates. In principle, any species could be used because the integral in $\dot{m}_m$ will be the same for all the species. However, it is best to choose one of the major species for this purpose since the numerical evaluation of this integral will not be as accurate if the species is a trace one.

To see how A58 and A59 automatically control the entrainment rates,

and thus the grid width, consider the situation in which the concentrations φ_{CC} and φ_{HH} differed from φ_C and φ_H by 1% of the total difference in φ across the grid. Then $(\varphi_{CC} - \varphi_C)/(\varphi_H - \varphi_C)$ and $(\varphi_H - \varphi_{HH})/(\varphi_H - \varphi_C)$ would equal 0.01 and from A58 and A59 we have $\dot{m}_C = \dot{m}_H = \dot{m}_m$. This would be the situation in the steady-state. Suppose this condition arose by accident at some early time in the calculation. Suppose also that as the calculation proceeded $(\varphi_{CC} - \varphi_C)/(\varphi_H - \varphi_C)$ increased. Then $\dot{m}_C$ would become somewhat greater than $\dot{m}_m$ and cold gas would be pulled into the flame front at an increased rate. If $\dot{m}_H$ had not changed, this would mean that the grid would start increasing in width. Increasing the grid width would tend to lower φ_{CC} for future times thus bringing $\dot{m}_C$ back to $\dot{m}_m$ thereby slowing down the rate of increase of grid size. Through the use of A58 and A59, the grid is automatically adjusted after every time step in such a way as to make $(\varphi_{CC} - \varphi_C)/(\varphi_H - \varphi_C)$ and $(\varphi_H - \varphi_{HH})/(\varphi_H - \varphi_C)$ tend to 0.01. This condition will finally be reached in the steady-state when all the φ cease to change with time.

The parameter ν in A58 and A59 is used to control the effect that departures in the 1% condition have on $\dot{m}_C$ and $\dot{m}_H$. For a stable calculation ν must be of the order of 0.1. Note that ν is purely a numerical device. We desire only to have some relation between $\dot{m}_{C/H}$ and the φ values such that $\dot{m}_{C/H} \rightarrow \dot{m}_m$ when the φ reach certain specified values. Other relationships which served a similar purpose could no doubt have been devised to perform the functions of A58 and A59.

It remains to show how the integral $\int (R_j/\rho M_j) d\omega$ is calculated. Recalling that $R_j/\rho M_j = \mathcal{S}_j$ (see page A21), we used the three point approximation A47 to give

$$\int_0^1 \mathcal{S}_j d\omega = \tfrac{1}{8}(3\boldsymbol{\Omega}\mathcal{S}_j + \boldsymbol{\Omega}_{+}\mathcal{S}_{j++} + \boldsymbol{\Omega}_{-}\mathcal{S}_{j--}). \qquad \text{(A61)}$$

The $\mathcal{S}_j$ values used in A61 are calculated from the values of φ at the end of each time step.

APPENDIX K

BOUNDARY CONDITIONS

Consider the finite-difference forms of the flame equations in final form. This is Eq. A43 derived in Appendix H.

$$\Phi_i = A_i^* \Phi_{i+1} + B_i^* \qquad (A43)$$

The subscripts refer to the grid points. There is one of these equations for each species and one for the enthalpy. To solve these equations we must assign a value to Φ_{N+2}, the grid point on the cold side of the flame. Furthermore, to calculate $B_3^* \equiv B_3 \Phi_2 + C_3$ we need to specify a value for Φ_2. This section is about how Φ_2 and Φ_{N+2} are to be chosen. The values will depend on the boundary conditions. We consider two types of flames; a) the freely propagating flame, and b) a flame stabilized on a burner.

a) Boundary conditions for a freely propagating flame

On the cold side of the species equations we set φ_{N+2} equal to their values in the cold incoming gas. On the hot side, we assume that the concentration gradient vanishes; thus we set $\varphi_2 = \varphi_3$. For the enthalpy, we calculate h_{N+2} from the temperature and composition of the cold gas mixture. Since the freely propagating flame is considered to be adiabatic, the value on the hot side is the same as that on the cold side. Thus we set $h_2 = h_{N+2}$.

b) Boundary conditions for a flame stabilized on a burner

The situation here is considerably more complicated than that occuring in the freely propagating flame. Consider the cold side of the grid. Locate the grid point N+2 at the surface of the burner. Figure A3 shows this portion of the grid along with the burner and the last control volume. The burner is considered to be made of some porous material which allows free passage of stable species, but not radicals. Cold reactant gases at a temperature T_∞ enter the porous

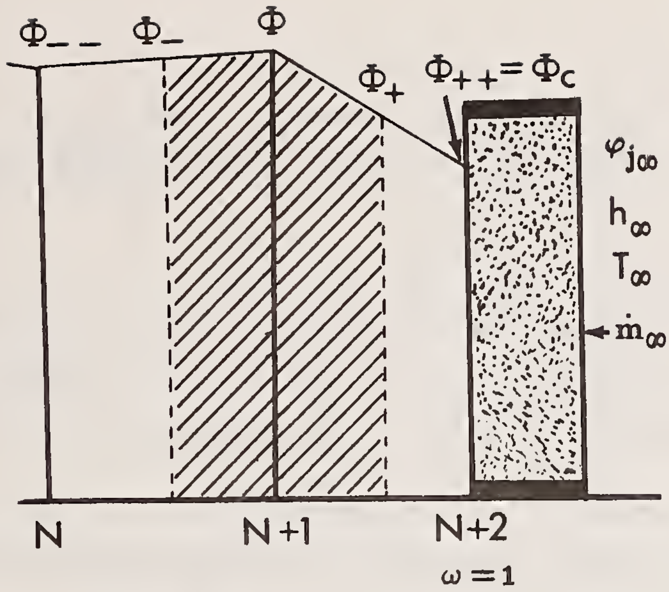


Figure A3.

material at a rate $\dot{m}_\infty$ (kg/m^2s). The composition and enthalpy of the mixture at this point is $\varphi_{j\infty}$ and h_∞. Since the porous plug is absorbing heat from the flame, there will be a temperature gradient within it. The temperature of the gas will thus increase to some extent as it passes through the plug. It is also conceivable that the concentrations φ_j on the warm side of the plug may be somewhat different from $\varphi_{j\infty}$ because of diffusion into and from the flame zone. Thus on the warm side of the plug which is located at the last grid point the gas properties have the values φ_{jC}, h_C, and T_C. These values are not known beforehand, but must be determined by solving the flame equations.

Consider first the boundary conditions for the species equations. Suppose Y_{jC} is the mass fraction of species j at the flame side of the plug. The mass flux of j at this point will be $(v + V_j)_C(\rho Y_j)_C$. This must be equal to the rate at which j emerges from the plug. This will be equal to the rate at which j enters the cold side of the plug, i.e., $(\rho v Y_j)_\infty$. As an additional source or sink for j we might also consider radicals diffusing out of the flame zone and combining on the burner surface. This effect is likely to be of importance, however, only for H atoms because of their large diffusion coefficient. To include this process in the model we define a surface distruction or production rate for the species j. Call this Θ_j. If Θ_j is in units of moles/m^2s, then $M_j\Theta_j$ will be the mass rate in kg/m^2s. For the particular case of H atoms recombining to give H_2, we can let $\Theta_H = -B_H(C_H)_C$, where $(C_H)_C$ is the concentration of H at the burner surface, and B_H is a rate constant for surface recombination. We will let negative values of Θ

correspond to destruction of the species. For the H_2 formed as a result of this, we would have $\Theta_{H_2} = \frac{1}{2}B_H(C_H)_C$. In general then, there will be an additional source of j equal to $-M_j\Theta_j$, and so we must have

$$(v + V_j)_C(\rho Y_J)_C = (\rho v Y_j)_\infty - M_j\Theta_j \tag{A62}$$

Since v is negative and denotes a flux from right to left along the y axis, $-M_j\Theta_j$ for $\Theta_j > 0$ also gives a negative flux which corresponds to production of j.

Since $\rho v = \dot{m}_\infty$ throughout the flame (note that $\dot{m}_\infty$ will be specified as part of the experimental conditions), A62 becomes

$$\dot{m}_\infty Y_{jC} + (\rho V_j Y_j)_C = \dot{m}_\infty Y_{j\infty} - M_j\Theta_j \tag{A63}$$

Inserting the expression for V_j (Eq. A5) into A63, we get

$$\left(\Delta_j \rho \frac{Y_j}{x_j} \frac{\partial x_j}{\partial y}\right)_C = \dot{m}_\infty(Y_{jC} - Y_{j\infty}) + M_j\Theta_j$$

Using the identity shown on page A9 and dividing the resulting equation by M_j gives

$$\left(\Delta_j \rho \frac{\partial \varphi_j}{\partial y} + \Delta_j \rho \frac{\varphi_j}{\langle M \rangle} \frac{\partial \langle M \rangle}{\partial y}\right)_C = \dot{m}_\infty(\varphi_{jC} - \varphi_{j\infty}) + M_j\Theta_j$$

If the gradient of $\langle M \rangle$ is neglected, this becomes

$$\left(\Delta_j \rho \frac{\partial \varphi_j}{\partial y}\right)_C = \dot{m}_\infty(\varphi_{jC} - \varphi_{j\infty}) + M_j\Theta_j$$

Transforming to the ω,t system with the relation $\partial/\partial y = (\rho/\eta)\partial/\partial\omega$ gives

$$\frac{1}{\eta}\left(\Delta_j \rho^2 \frac{\partial \varphi_j}{\partial \omega}\right)_C = \dot{m}_\infty(\varphi_{jC} - \varphi_{j\infty}) + M_j\Theta_j \tag{A64}$$

Since the φ profile is assumed to be linear between grid points (see Fig. A3), the concentration gradient at point C, i.e., i=N+2, will be the same as that at the mid-point between i=N+1 and N+2. Thus

$$\left(\frac{\partial \varphi_j}{\partial \omega}\right)_C = \left(\frac{\partial \varphi_j}{\partial \omega}\right)_+ = \frac{\varphi_{j++} - \varphi_j}{\omega_{++} - \omega} = \frac{\varphi_{jN+2} - \varphi_{jN+1}}{\omega_{N+2} - \omega_{N+1}} \tag{A65}$$

Inserting this into A64 and solving for $\varphi_{jN+2} = \varphi_{jC}$, we get

$$\varphi_{jN+2} = \left[\frac{\Delta_j\rho^2}{\eta(\omega_{N+2}-\omega_{N+1})}\varphi_{jN+1} - \dot{m}_\infty\varphi_{j\infty} + M_j\Theta_j\right]$$

$$\div\left[\frac{\Delta_j\rho^2}{\eta(\omega_{N+2}-\omega_{N+1})} - \dot{m}_\infty\right] \qquad \text{(A66)}$$

At the hot boundary we assume $\partial\varphi/\partial\omega = 0$, and thus we set $\varphi_2 = \varphi_3$ as in the freely propagating flame.

Next we consider the boundary conditions on the energy equation. The net flux of energy at the burner surface will be the sum of that caused by the mass flow of the reactant mixture, the flow caused by diffusion of species to and from the surface, and the flow arising from a temperature gradient in the gas at the surface. Let $(F_h)_C$ be this net energy flux at the burner.

$$(F_h)_C = -(\lambda\frac{\partial T}{\partial y})_C + \sum_j\left[(v + V_j)\rho Y_j h_j\right]_C$$

$$= -(\lambda\frac{\partial T}{\partial y})_C + \dot{m}_\infty h_C + \sum_j(V_j\rho Y_j h_j)_C \qquad \text{(A67)}$$

Introducing the expression for V_j and the identity on page A9, neglecting the gradient of $\langle M\rangle$, converting to φ_j concentration units, and going to the ω,t system, we get

$$(F_h)_C = -\frac{1}{\eta}\left(\frac{\lambda\rho}{C_p}\frac{\partial h}{\partial\omega}\right)_C + \dot{m}_\infty h_C - \frac{1}{\eta}\sum_j\left[\left\{\Delta_j\rho^2 - \frac{\lambda\rho}{C_p}\right\}h_j^*\frac{\partial\varphi_j}{\partial\omega}\right]_C \qquad \text{(A68)}$$

Since this is the net flux of energy into the cold side of the flame front, it must be equal to the net flux out of the flame on the hot side. Because we assume that $\partial T/\partial y$ and $\partial\varphi_j/\partial y = 0$ there, this flux out is $\dot{m}_\infty h_H$ where h_H is the enthalpy of the hot gas mixture. Assuming a linear variation of h between grid points gives

$$\left(\frac{\partial h}{\partial\omega}\right)_C = \left(\frac{\partial h}{\partial\omega}\right)_+ = \frac{h_{N+2} - h_{N+1}}{\omega_{N+2} - \omega_{N+1}} = \frac{h_C - h_{N+1}}{\omega_C - \omega_{N+1}}$$

Inserting this into A68, and setting $(F_h)_C = \dot{m}_\infty h_H$, gives

$$\dot{m}_\infty h_H = -\frac{1}{\eta}\left\{\frac{\lambda\rho}{C_p}\right\}_C\left\{\frac{h_C - h_{N+1}}{\omega_C - \omega_{N+1}}\right\} + \dot{m}_\infty h_C - \frac{1}{\eta}\sum_j\left[\left\{\Delta_j\rho^2 - \frac{\lambda\rho}{C_p}\right\}h_j^* \frac{\partial\varphi_j}{\partial\omega}\right]_C \quad \text{(A69)}$$

This equation can be solved to give h_{N+2} ($=h_C$) in terms of h_{N+1} and h_H, and other known quantities. The enthalpy at the hot side can be calculated from the measured final flame temperature and its composition. From h_C and the calculated composition $(\varphi_j)_C$ at the burner surface, we can also evaluate T_C.

APPENDIX L.

TRANSFORMATION FROM THE ω,t COORDINATE SYSTEM BACK TO THE y,t SYSTEM

The steady-state solution of the species and energy equations A22 and A23 in the ω,t coordinate system yields values of φ_j and h as functions of ω, where $0 \leq \omega \leq 1$. If we want to compare these profiles with experimental ones we must express φ_j and h in terms of y, the laboratory spatial variable.

The variable φ is a function of ψ and t, and ψ is a function of y and t; thus,

$$\omega = \omega(\psi,t) = \omega(\psi(y,t),t)$$

A small change in ψ can be expressed as

$$d\psi = \frac{\partial\psi}{\partial y} dy + \frac{\partial\psi}{\partial t} dt = \rho dy - \rho v dt$$

Here we have used A18, the expressions defining ψ. A small change in ω can likewise be written

$$d\omega = \frac{\partial\omega}{\partial\psi} d\psi + \frac{\partial\omega}{\partial t} dt = \frac{\partial\omega}{\partial\psi}(\rho dy - \rho v dt) + \frac{\partial\omega}{\partial t} dt$$

$$= \frac{\partial\omega}{\partial\psi} \rho dy + \left(-\rho v \frac{\partial\omega}{\partial\psi} + \frac{\partial\omega}{\partial t}\right)dt$$

Using the facts that $\frac{\partial\omega}{\partial\psi} = \frac{1}{\eta}$ and $\frac{\partial\omega}{\partial t} = \frac{a + b\omega}{\eta}$, we get

$$d\omega = \frac{\rho}{\eta} dy + \left(-\frac{\rho v}{\eta} + \frac{a + b\omega}{\eta}\right) dt$$

In the steady state $b \to 0$ and $a \to \dot{m}_C = \rho v$. Therefore $d\omega \to \frac{\rho}{\eta} dy$.

Solving this for dy and integrating from $\omega = 0$ (the hot side) to some arbitrary ω value, we get

$$\int_{y_H}^{y} dy = \eta \int_0^{\omega} \frac{d\omega}{\rho} = y - y_H$$

In the present calculation, we arbitrarily set $y_H = 0$. Thus we can calculate the value of y which corresponds to a particular value of ω by evaluating the integral

$$y = \eta \int_0^{\omega} \frac{d\omega}{\rho}$$

This integral is approximated in the computer program by the summation

$$y_j = 2\eta \sum_{i=2}^{j} \frac{(\omega_{i+1} - \omega_i)}{\rho_{i+1} - \rho_i}, \tag{A70}$$

where the subscripts refer to the grid points.

APPENDIX M.

CALCULATION OF THERMAL AND TRANSPORT PROPERTIES FOR PURE SPECIES

An expression for the effective diffusion coefficient Δ_j of a particular species in the flame has been derived in Appendix C (eq.A9). This gives Δ_j in terms of the concentrations of the various species and the binary diffusion coefficients for all possible pairs of species. In practice, the summation in the denominator of A9 need not be evaluated over all the species since x_i, the mole fraction, is large only for the major species. Thus, we must calculate only the binary diffusion coefficients between major species, and between major and minor species. Those coefficients involving pairs of minor species need not be evaluated. In Appendix D we showed how the thermal conductivity of the flame was calculated. The expression used contains the thermal conductivities of the pure species, and also their heat capacities. The heat capacities of the pure species are also needed to calculate C_p, the average heat capacity of the flame. This quantity occurs in the energy equation. For the energy equation, we also require values for the enthalpies of the pure species. Thus, we must evaluate the following properties involving the pure species (or pairs of species in the diffusion case):

a. Binary diffusion coefficients D_{ij}.

b. Thermal conductivities λ_j.

c. Heat capacities C_{pj}.

d. Enthalpies h_j (or $h_j^* = h_j M_j$).

a) <u>Binary diffusion coefficients</u>

For the diffusion coefficients, we used Lennard-Jones potentials. This potential function has two parameters σ_j and ϵ_j/k. σ_j gives the distance in Å where the potential goes from repulsive the attractive, and ϵ_j/k in $^{\circ}$K gives the depth of the potential well. σ_j and ϵ_j refer the the interaction between two molecules of the same species. For the diffusion coefficients, we require analogous parameters for the interaction between molecules of different species. If σ_i, σ_j and ϵ_i, ϵ_j are the parameters for the species i and j, we take as the

parameters for the interaction between i and j to be

$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j)$ and $\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}$.

The formula for the binary diffusion coefficient is

$$D_{ij} = \frac{1.86 \times 10^{-7} \left[(M_i + M_j)/M_i M_j\right]^{\frac{1}{2}} T^{1.5}}{P\sigma_{ij}^2 \quad \Omega_{ij}^{(1,1)*}} \tag{A71}$$

where P is the pressure in atmospheres, M_i and M_j are the molecular weights, and $\Omega_{ij}^{(1,1)*}$ is a function of a reduced temperature $T^* = Tk/\epsilon_{ij}$. To a good approximation this function is proportional to T^a where a takes on two constant values, one in the range $T \geqq 3\epsilon_{ij}/k$ and another for $T < 3\epsilon_{ij}/k$. We have calculated D_{ij} from either of the following formulas depending on whether the temperature was greater of less than $3\epsilon_{ij}/k$.

$$D_{ij} = \frac{1.66 \times 10^{-7} \left[(M_i + M_j)/M_i M_j\right]^{\frac{1}{2}} T^{1.67}}{P\sigma_{ij}^2 \, (\epsilon_{ij}/k)^{0.17}} \qquad T \geqq 3\epsilon_{ij}/k \tag{A72}$$

$$D_{ij} = \frac{1.26 \times 10^{-7} \left[(M_i + M_j)/M_i M_j\right]^{\frac{1}{2}} T^{1.94}}{P\sigma_{ij}^2 \, (\epsilon_{ij}/k)^{0.44}} \qquad T < 3\epsilon_{ij}/k \tag{A73}$$

These formulas will give D_{ij} in units of m^2/s.

b) Thermal conductivities

Lennard-Jones potentials can also be used to calculate the thermal conductivities of the pure species. The formula for the thermal conductivity is similar to that for the diffusion coefficients. In a like manner we have found that it is possible to express it approximately in terms of a constant power of T, where the appropriate value of the exponent depends on whether the temperature is above or below $3\epsilon_j/k$. The formulas obtained were,

$$\lambda_j = \frac{1.604 \times 10^{-4} M_j^{-\frac{1}{2}} T^{0.67}}{\sigma_j^2 \, (\epsilon_j/k)^{0.17}} \qquad T \geqq 3\epsilon_j/k \tag{A74}$$

$$\lambda_j = \frac{1.231\ 10^{-4} M_j^{-\frac{1}{2}}\ T^{0.94}}{\sigma_j^2\ (\epsilon_j/k)^{0.44}} \qquad T < 3\epsilon_j/k \qquad \text{(A75)}$$

These two formulas give the thermal conductivities in units of cal/cm-deg-s. To be usable in the program they would have to be converted to J/m-deg-s units.

In the present version of the program, experimental thermal conductivity data were used for the major species in the H_2-O_2 flame. The data were fit by a least squares calculation to a power series in T and the least squares coefficients were used in the program to calculate the λ_j values.

c) Heat Capacities

Heat capacity data from the JANAF Tables were fit by least squares calculations to a power series in T. Terms up to the second power were used. Thus, heat capacities were calculated from the formula

$$C_{pj} = d_j + e_j T + f_j T^2 \qquad \text{(A76)}$$

where d_j, e_j, and f_j have values such that the C_{pj} are in units of J/kg-deg.

d) Enthalpies

The enthalpy of a species in J/kg is given by the formula

$$h_j = h_j^R + \int_{T_R}^{T} C_{pj} dT \qquad \text{(A77)}$$

where h_j^R is the enthalpy at the reference temperature T_R.
Substituting A76 into A77, we get

$$h_j = (h_j^R - d_j T_R - \tfrac{1}{2} e_j T_R^2 - \tfrac{1}{3} f_j T_R^3) + d_j T + \tfrac{1}{2} e_j T^2 + \tfrac{1}{3} f_j T^3 \qquad \text{(A78a)}$$

$$= h_j^o + d_j T + \tfrac{1}{2} e_j T^2 + \tfrac{1}{3} f_j T^3 \qquad \text{(A78b)}$$

This formula was used in the program to calculate h_j values. Values of h_j^o were calculated from h_j^R values taken from the JANAF Tables, the T_R values, and the heat capacity coefficients d_j, e_j, and f_j.

APPENDIX N.
INITIAL PROFILES

The mixture of cold gases through which the flame will propagate is naturally thermodynamically unstable. Nevertheless, in most systems of practical interest the rate of approach to equilibrium at low temperatures is negligibly slow. This is because the free radical reactions which destroy the reactants normally have large activation energies. There are two ways to speed up these reactions. One can raise the temperature of the system or introduce a sufficiently high concentration of the necessary radicals. For a flame steadily propagating into a cold unstable gas mixture, both of these processes occur. The cold gases receive heat by thermal conduction and radicals by molecular diffusion from the flame region. To solve the time dependent flame equations, one must begin with the system in such a state that the radical reactions are proceeding at an appreciable rate. Furthermore, it is desirable to start the calculation with concentration and energy profiles which are similar to those expected in the final steady state to minimize the number of steps required for the integration.

These goals were satisfied by starting the calculations with S-shaped profiles for temperature and for the concentration of the major species. These profiles were obtained by using the following function of ω, the spatial variable in the Spalding coordinate system:

$$P_d(\omega) = 10\omega^3 - 15\omega^4 + 6\omega^5 \qquad \text{(A79)}$$

This has the boundary values, $P_d(0) = 0$, and $P_d(1) = 1$. Since $\omega = 0$ corresponds to the hot side of the flame, A79 represents a decay profile. Formation profiles were obtained from the function

$$P_f(\omega) = 10(1-\omega)^3 - 15(1-\omega)^4 + 6(1-\omega)^5 \qquad \text{(A80)}$$

At the start of the calculation, we specify the mole fractions of a major species on the hot and cold sides of the flame. The former is calculated by assuming that the overall reaction has gone to completion. Also specified is whether the species decays or grows as we go from the cold to the hot side of the flame.

For example, suppose the mole fraction of species j is x_{jC} and x_{jH} for $\omega = 1$ and 0, and also suppose that $x_{jC} > x_{jH}$, i.e., we have a decay profile; then

$$x_j(\omega) = x_{jH} + (x_{jC} - x_{jH})(10\omega^3 - 15\omega^4 + 6\omega^5)$$

To calculate an initial temperature profile, we first calculate the enthalpy on the cold side from the cold side temperature T_C with the expression,

$$h_C = \sum_{j=j_R+1}^{n} M_j\ \varphi_{jC}(h_j^o + d_jT_C + \tfrac{1}{2}e_jT_C^2 + \tfrac{1}{3}f_jT_C^3) \qquad \text{(A81)}$$

φ_{jC} is the concentration of j on the cold side in units of mole/kg. We then set the enthalpy on the hot side equal to that on the cold side. (Note that j_R is the index of the last radical species; i.e., we sum over the major species only.) The expression for the hot side enthalpy is

$$h_H = h_C = \sum_{j=j_R+1}^{n} M_j\ \varphi_{jH}(h_j^o + d_jT_H + \tfrac{1}{2}e_jT_H^2 + \tfrac{1}{3}f_jT_H^3) \qquad \text{(A82)}$$

$$= A + BT_H + \tfrac{1}{2}CT_H^2 + \tfrac{1}{3}DT_H^3$$

where

$$A = \sum_{j=j_R+1}^{n} M_j\ \varphi_{jH}h_j^o$$

$$B = \sum_{j=j_R+1}^{n} M_j\ \varphi_{jH}d_j$$

$$C = \sum_{j=j_R+1}^{n} M_j\ \varphi_{jH}e_j$$

$$D = \sum_{j=j_R+1}^{n} M_j\ \varphi_{jH}f_j$$

We want to calculate T_H in terms of h_H. This was done by finding the appropriate root of the equation

$$F(T_H) = A + BT_H + \tfrac{1}{2}CT_H^2 + \tfrac{1}{3}DT_H^3 - h_H = 0$$

A good first approximation to the desired root can be gotten by neglecting the non-linear portion of A82. This gives $T_H^{(1)} = (h_H - A)/B$.

The Newton-Raphson method was used to get a better approximation. The second approximation is

$$T_H^{(2)} = T_H^{(1)} - F(T_H^{(1)})/F'(T_H^{(1)})$$

$$= T_H^{(1)} + \left\{h_H - (A + BT_H^{(1)} + \tfrac{1}{2}CT_H^{(1)2} + \tfrac{1}{3}DT_H^{(1)3})\right\}/(B + CT_H^{(1)} + DT_H^{(1)^2})$$

In practice, $T_H^{(3)}$ was a sufficiently accurate value for T_H.

The Newton-Raphson method was also used in the main part of the program to calculate the temperature from the enthalpy value at a particular grid point.

An S-shaped starting temperature profile was calculated from the formula

$$T(\omega) = T_C + (T_H - T_C)(10(1-\omega)^3 - 15(1-\omega)^4 + 6(1-\omega)^5)$$

For the enthalpy, a constant starting profile $h(\omega) = h_C$ was used. It should be noted that this profile is not consistent with the starting temperature profile. The enthalpy profile should be calculated from the temperature and concentration starting profiles. It would be worthwhile to examine what effect, if any, the starting enthalpy profile has on the course of the calculation.

The radical concentrations were set to zero on the cold boundary and remained so throughout the calculation, since the cold boundary concentrations of all species are never modified. Throughout the rest of the flame they initially were set to some constant value comparable to their average expected steady-state concentrations. For flames like O_3 decomposition and $H_2 - Br_2$ the initial radical concentrations can be set to zero throughout the flame zone. This is possible because the radical generating reactions like $Br_2 + M \rightleftarrows 2Br + M$ proceed rapidly on the hot side of the flame. For the $H_2 - O_2$ flame however, such reactions are too slow for cool flames to initially generate enough radicals. The flame dies out before the radical concentrations become high enough to sustain the flame. Thus, one must begin the calculation with non-zero radical concentrations.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 76-994	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE A Numerical Solution of the Time Dependent Partial Differential Equations which Describes a one-dimensional, Laminar, Premixed Flame	5. Publication Date January 1976 6. Performing Organization Code
7. AUTHOR(S) R. L. Brown	8. Performing Organ. Report No. NBSIR 76-994
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234	10. Project/Task/Work Unit No. 3162675 11. Contract/Grant No.
12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)	13. Type of Report & Period Covered Final 14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The set of time dependent, parabolic differential equations, which describe the physical and chemical processes in a one-dimensional, laminar, premixed flame is solved by adapting a solution procedure originally developed to solve the two-dimensional steady state boundary layer equations. The flame equations are integrated by an implicit method until the steady state is reached. This corresponds to a flame propagating steadily through a mixture of combustible gases. By a suitable choice of boundary conditions, it is also possible to model a flame which is stabilized on a burner. Solution of the flame equations yields the concentration profiles of the different chemical species as well as the temperature profile. From these one can also calculate the production rates of each species, the rate of each chemical reaction, and the heat release rate at each point in the flame. The velocity of the freely propagating flame can be calculated from the integrals over the whole flame zone of any of the species production rates. The model incorporates realistic thermodynamic data and transport property data that are functions of both temperature and concentration. A complete documentation of the computerprogram which accomplishes the integration is presented.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)* Combustion; differential equations; flame chemistry; flame modeling; laminar flames; numerical solution of flame equations

18. AVAILABILITY	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
[X] Unlimited [ ] For Official Distribution. Do Not Release to NTIS [ ] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13 [X] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151	UNCLASSIFIED 20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	175 22. Price $6.75

USCOMM-DC 29042-P74